



**IEEE Standard for
Local and metropolitan area networks**

**Part 20: Air Interface for Mobile
Broadband Wireless Access Systems
Supporting Vehicular Mobility—
Physical and Media Access Control Layer
Specification**

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997, USA
29 August 2008

IEEE Std 802.20™-2008

802.20™

IEEE Standard for
Local and metropolitan area networks

**Part 20: Air Interface for Mobile
Broadband Wireless Access Systems
Supporting Vehicular Mobility—
Physical and Media Access Control Layer
Specification**

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 12 June 2008

IEEE-SA Standards Board

The IEEE would like to acknowledge with thanks that permission has been granted by the Alliance for Telecommunications Industry Solutions (ATIS), on behalf of its Wireless Technologies and Systems Committee's (WTSC) Wireless Wideband Internet Access (WWINA), to the IEEE 802.20 Mobile Broadband Wireless Access (MBWA) Working Group to include specific references to the ATIS standard, ATIS High Capacity—Spatial Division Multiple Access (HC-SDMA) Standard (ATIS-0700004-2005), within IEEE Std 802.20.

Grateful acknowledgement is also made to the Telecommunications Industry Association (TIA) for granting permission the IEEE to use the below listed TIA standards for world rights distribution, with additional permission to modify and reprint in all future revisions and editions of IEEE Std 802.20 in all media known or hereinafter known.

TIA-1121.000—Overview for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.001—Physical Layer for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.002—MAC Layer for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.003—Radio Link Layer for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.004—Application Layer for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.005—Security Functions for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.006—Connection Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.007—Session Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.008—Route Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification

TIA-1121.009—Broadcast-Multicast Upper Layers for Ultra Mobile Broadband (UMB) Air Interface Specification

Abstract: The technical requirements of this standard form a compatibility standard for mobile broadband wireless access systems. The standard ensures that a compliant access terminal (AT) or user terminal (UT) can obtain service through any access node (AN) or base station (BS) conforming to properly selected modes of this standard, consistent with equipment and operator requirements, thus providing a framework for the rapid development of cost-effective, multivendor mobile broadband wireless access systems. This compatibility standard is targeted for use in a wide variety of licensed frequency bands and regulatory environments.

This standard provides for future standardized extension capabilities. The architecture defined by this standard permits such expansion without the loss of backward compatibility.

Keywords: access node, access terminal, base station, broadband, LAN, local area networks, MAN, metropolitan area networks, user terminal, vehicular mobility, wireless access

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2008 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 29 August 2008. Printed in the United States of America.

IEEE and 802 are registered trademarks in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 973-07381-5766-5 STD95805

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS.**”

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgement in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be submitted to the following address:

Secretary, IEEE-SA Standards Board

445 Hoes Lane

Piscataway, NJ 08854

USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

This introduction is not part of IEEE Std 802.20-2008, IEEE Standard for Local and metropolitan area networks—Part 20: Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility—Physical and Media Access Control Layer Specification.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association website at <http://ieeexplore.ieee.org/xpl/standards.jsp>, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at <http://standards.ieee.org>.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/updates/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/interp/index.html>.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the Mobile Broad-band Wireless Access Working Group had the following membership:

Arnold Greenspan, *Chair*
Mark Klerer, *Vice Chair*
James Mollenauer, *Vice Chair*
Donald Gillies, *Recording Secretary*

Gwen Barriac
Moussa Bavafa
Jaffrey Bernstein
Nancy S. Bravin
Petru Budianu
Chris Bussey
Radhakrishna Canchi
Peter Carson
Harold Castel
Chris Dean
Doug Dunn
Mark Epstein
Peretz Feder
Graham C. Freeland
Daniel Garcia-Alis
Deepshikha Garg
Eladio Gomes
Dhananjay Gore
Svetlana Gorodetsky
Alexei Gorokhov
Radhika Gowaikar
Victor Hou
Kazuyoshi Iimuro
Kazuhito Ishida
Allan Jette
Tingfang Ji
Dennis Jones
Tamer Kadous

Amit Kalhan
Majid Khademi
Aamod Khandekar
Farrokh Khatibi
Shigeru Kimura
Minako Kitahara
Douglas Knisely
Tom Kolze
Changhoi Koo
Lisa Lawrence
Heesoo Lee
Irina Martynov
Marty Martynov
Anthony McMahon
Donald McMillan
Max Miyazano
Kazuhiro Murakami
Mullagru Naidu
Kenichi Nakamura
Tetsuya Nakamura
Shinji Nakano
Nha Nguyen
Fred Novick
Francis O'Brien
Hiroshi Oguma
Eugen Pfann
Riku Pirhonen
Sebastien Poisson

Rajat Prakash
Rob Preece
James H. Ragsdale
Ajay Rajkumar
Hemanth Sampath
Masato Sano
Shigenobu Sasaki
Judy Shields
David Shively
Roshni Srinivasian
Doug Staver
Richard Stuby
Mark Suh
Jing Sun
Valentin Surcobe
Tomohiro Suzuki
Harris Teague
Lai-King Anna Tee
James Tomcik
Jerry Upton
Jaun Carlos Valls
Rajiv Vijayan
Silvia Vivanco
Michael Wang
Thomas Wasilewski
Joanne Wilson
Rao Yallapragada
Michael Youssefmir
Masaaki Yuza

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Toru Aihara	Randall Groves	Satoshi Obara
Thomas Alexander	C. Guy	Francis O'Brien
Danilo Antonelli	Gavin Horn	Hiroshi Oguma
Matthew Ball	Victor Hou	Chris Osterloh
Phillip Barber	John Humbert	Satoshi Oyama
John Barr	Tetsushi Ikegami	Stephen Palm
Gennaro Boggia	Atsushi Ito	William Pechey
Nancy S. Bravin	Raj Jain	Eugen Pfann
William Byrd	Tony Jeffree	Riku Pirhonen
Sean Cai	Alan Jette	Subburajan Ponnuswamy
Peter J. Calderon	Tingfang Ji	Michael Probasco
Radhakrishna Canchi	Stuart J. Kerry	Jose Puthenkulam
Juan Carreon	Farrokh Khatibi	James H. Ragsdale
Yung-Mu Chen	Brian Kiernan	Maximilian Riegel
Keith Chow	Eunkyung Kim	Robert Robinson
Joseph Cleveland	Yongho Kim	Herbert Ruck
Todor Cooklev	Mark Klerer	Shigenobu Sasaki
Russell Dietz	Tom Kolze	Thomas Starai
Carlo Donati	Jonathan Labs	Carl Stevenson
Carl Eklund	Jeremy Landt	Walter Struppler
Marc Emmelmann	Heesoo Lee	Alourdes Sully
Mark Epstein	Jan-Ray Liao	Jing Sun
Bernard Eydt	G. Luri	Masato Tanahashi
Prince Francis	Roger Marks	John Terry
Avraham Freedman	Peter Martini	Geoffrey Thompson
Graham C. Freeland	Gary Michel	James Tomcik
Devon Gayle	Apurva Mody	Jerry Upton
Pieter-Paul Giesberts	Avygdor Moise	Prabodh Varshney
James Gilb	Jose Morales	Stanley Wang
Donald Gillies	Ronald G. Murias	Fujio Watanabe
Mariana Goldhamer	Ayman Naguib	David Willow
Sergiu Goma	Juichi Nakada	Joanne Wilson
Radhika Gowaikar	Hiroyuki Nakase	Michael Youssefmir
Arnold Greenspan	Michael S. Newman	Oren Yuen

When the IEEE-SA Standards Board approved this standard on 12 June 2008, it had the following membership:

Robert M. Grow, *Chair*
Thomas Prevost, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Victor Berman
Richard DeBlasio
Andy Drozd
Mark Epstein
Alexander Gelman
William Goldbach
Arnie Greenspan
Ken Hanus

Jim Hughes
Richard Hulett
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Glenn Parsons

Ron Petersen
Chuck Powers
Narayanan Ramachandran
Jon Walter Rosdahl
Anne-Marie Sahazizian
Malcolm Thaden
Howard Wolfman
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Michael H. Kelley, *NIST Representative*

Michelle Turner
IEEE Standards Program Manager, Document Development

Michael D. Kipness
IEEE Standards Program Manager, Technical Program Development

Contents

1.	Scope.....	1
2.	Normative references	2
3.	Definitions	4
4.	Abbreviations and acronyms	19
5.	General introduction	21
	5.1 General.....	21
	5.2 Modes of the standard.....	21
	5.3 Requirements language.....	21
	5.4 Wideband mode overview	21
	5.5 System overview of 625k-MC (625kiloHertz-spaced MultiCarrier) Mode	38
6.	Services Sublayer.....	41
	6.1 Introduction.....	41
	6.2 Basic Signaling Protocol.....	41
	6.3 Basic Inter-Route Tunneling Protocol	45
	6.4 Basic Robust Header Compression Support Protocol	53
	6.5 Basic EAP Support Protocol.....	58
7.	Radio Link Sublayer	62
	7.1 Introduction.....	62
	7.2 Basic QOS Management Protocol	63
	7.3 Basic Radio Link Protocol.....	94
	7.4 Basic Stream Protocol.....	131
	7.5 Basic Route Protocol	135
8.	Lower MAC Sublayer.....	146
	8.1 General.....	146
	8.2 Basic Packet Consolidation Protocol	159
	8.3 Basic Superframe Preamble Mac Protocol	166
	8.4 Basic Access Channel MAC Protocol	176
	8.5 Basic Forward Link Control Segment MAC Protocol.....	189
	8.6 Basic Forward Traffic Channel MAC Protocol.....	218
	8.7 Basic Reverse Control Channel MAC Protocol	291
	8.8 Basic Reverse Traffic Channel MAC Protocol.....	346
9.	Physical Layer.....	389
	9.1 Basic Physical Layer Protocol	389
	9.2 General.....	390
	9.3 Requirements for access terminal operation	496
	9.4 Access network requirements	521
	9.5 Broadcast Multicast Services (BCMCS)	569

10.	Security functions	586
10.1	Introduction	586
10.2	AES Ciphering Protocol.....	586
10.3	Basic Message Integrity Protocol.....	595
10.4	Basic Key Exchange Protocol	603
11.	Connection Control Sublayer.....	614
11.1	Introduction	614
11.2	Basic Air Link Management Protocol.....	615
11.3	Basic Initialization State Protocol	640
11.4	Basic Idle State Protocol.....	644
11.5	Basic Connected State Protocol.....	659
11.6	Overhead Messages Protocol.....	681
11.7	Basic Active Set Management Protocol	714
12.	Session Control Plane	754
12.1	Introduction	754
12.2	Basic Session Control Protocol.....	755
13.	Route Control Plane.....	798
13.1	Introduction	798
13.2	Basic Route Control Protocol.....	798
14.	Broadcast support	825
14.1	Introduction	825
14.2	Basic Broadcast Protocol Suite	825
15.	Common procedures and data structures	877
15.1	Common procedures	877
15.2	Common data structures.....	880
16.	Assigned names and numbers	889
16.1	Protocol types and subtypes	889
16.2	ANID, SectorID, and UATI provisioning.....	892
17.	MAC and PHY MIB	894
17.1	Overview	894
17.2	MIB structure	894
17.3	Definition	894
18.	625k-MC spectral layout terminology and requirements	935
19.	625k-MC slot and frame structure	936
19.1	Overview	936
19.2	RF channel and frame structure.....	936

19.3	Burst formats	936
20.	625k-MC modulation and channel coding.....	937
20.1	625k-MC modulation and channel coding overview	937
20.2	Standard modulation and coding.....	937
20.3	Broadcast channel modulation and coding	944
21.	625k-MC user terminal radio transmission and reception.....	945
21.1	625k-MC user terminal nominal output power for modulation formats	945
21.2	625k-MC UT modulation accuracy for modulation formats.....	945
21.3	625k-MC UT reference sensitivity level.....	945
21.4	625k-MC UT receiver SINR estimation accuracy	946
21.5	625k-MC UT receiver adjacent channel selectivity	946
22.	625k-MC base station radio transmission and reception	947
22.1	625k-MC base station transmitter modulation accuracy	947
22.2	625k-MC base station receiver reference sensitivity level.....	947
22.3	625k-MC BS receiver SINR estimation accuracy.....	947
23.	625k-MC L2 MAC Protocol Sublayer Specification	948
23.1	Logical channels.....	948
23.2	625k-MC minimized RMU header.....	950
24.	625k-MC L2 RLC Protocol Sublayer Specification.....	953
24.1	625k-MC AM RMU	953
24.2	625k-MC transmit procedure	953
24.3	Receive procedure	955
25.	625k-MC L3 Protocol Specification.....	957
26.	625k-MC Protocol layer primitives (informative).....	958
26.1	Interface list	958
26.2	Individual interfaces	958
27.	625k-MC QoS Enhancements	965
27.1	Classes of services.....	965
27.2	Session QoS information exchange procedures	965
27.3	QoS priority.....	966
28.	625k-MC Broadcast and Multicast Service (BCMCS) Support Enhancement	968
28.1	Overview	968
28.2	Broadcast service	968
28.3	Multicast service	969
29.	625k-MC Privacy and Authentication Enhancement	975

29.1	Overview	975
29.2	625k-MC Handshake and BS Authentication Protocol, i-HAP	975
29.3	625k-MC Terminal Authentication Protocol, i-TAP	976
29.4	625k-MC Secure Communications Protocol, i-SEC	976
30.	625k-MC Sleep Mode Control Protocol	981
31.	625k-MC OA & M Radio Network Quality Monitor and Control Enhancement	982
31.1	625k-MC Mode MIB	982
Annex A (informative) 625K-MC Glossary of Technical Terms		1038
Annex B (informative) Bibliography		1039

IEEE Standard for
Local and metropolitan area networks

Part 20: Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility— Physical and Media Access Control Layer Specification

IMPORTANT NOTICE: This standard is not intended to assure safety, security, health, or environmental protection in all circumstances. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

1. Scope

This standard specifies the physical and medium access control layers of an air interface for interoperable mobile broadband wireless access systems, operating in licensed bands below 3.5 GHz. The system is optimized for IP-data transport, with peak data rates per user in excess of 1 Mbps.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

3GPP2 C.R1001-F v1.0, Administration of Parameter Value Assignments for cdma2000TM Spread Spectrum Standards.¹

3GPP2 C.S0001, Introduction to cdma2000 Standards for Spread Spectrum Systems.

3GPP2 C.S0010-C v2.0, Recommended Minimum Performance Standards for cdma2000 Spread Spectrum Base Stations, March 2006.

3GPP2 C.S0011-C, Recommended Minimum Performance Standards for cdma2000 Spread Spectrum Mobile Stations, March 2006.

3GPP2 C.S0024, cdma2000 High Rate Packet Data Air Interface Specification.

3GPP2 C.S0057-B, Band Class Specification for cdma2000 Spread Spectrum Systems.

3GPP2 S.S0055A v2.0, Enhanced Cryptographic Algorithms.

3GPP2 S.S0078, Common Security Algorithms.

ATIS-0700004.2005, High Capacity-Spatial Division Multiple Access (HC-SDMA) Radio Interface Standard, September 2005.²

IEEE Std 802.1Q™, IEEE Standard for Local and Metropolitan Area Networks: 2 Virtual Bridged Local Area Networks.^{3, 4}

IEEE Std 802.2™, IEEE Standard for Local Area Networks—Logical link control.

IEEE Std 802.3™, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

IETF RFC 791, Internet Protocol.⁵

IETF RFC 2210, The Use of RSVP with IETF Integrated Services.

IETF RFC 2216, Network Element Service Specification Template, S. Shenker, J. Wroclawski, September 1987.

IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification.

¹3GPP2 publications are available at <http://www.3gpp2.org/Public>.

²© Copyright 2006 by the Alliance for Telecommunications Industry Solutions, Inc. ATIS publications are available from the ATIS Document Store at <https://www.atis.org/docstore/default.aspx>.

³IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (<http://standards.ieee.org/>).

⁴The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

⁵IETF publications are available from the Internet Engineering Task Force at <http://www.ietf.org/>.

IETF RFC 2578, Structure of Management Information Version 2 (SMIv2), April 1999.

IETF RFC 2579, Textual Conventions for SMIv2, April 1999.

IETF RFC 2580, Conformance Statements for SMIv2, April 1999.

IETF RFC 3056, Connection of IPv6 Domains via IPv4 Clouds, February 2001.

IETF RFC 3095, Robust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed.

IETF RFC 3748, Extensible Authentication Protocol, June 2004.

IETF RFC 4493, *The AES-CMAC Algorithm*, Song, JH., Poovendran, R., Lee, J., and Iwata, T., June 2006.

ITU-T Recommendation E.212, Identification Plan for Land Mobile Stations, 1988.⁶

NIST, Special Publication 800-38B, *Recommendation for Block Cipher Modes of Operation: The CMAC Method for Authentication*.

⁶ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (<http://www.itu.int/>).

3. Definitions

For the purposes of this document, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standard Terms* [B1]⁷ should be referenced for terms not defined in this clause.

3.1 access network: Access network is a collection of access nodes (ANs).

3.2 access node (AN): The physical device that implements the MAC and PHY of one or more sectors in the network.

3.3 access probe: A sequence of signaling transmitted by the access terminal on the reverse access channel to establish a connection to the access network.

3.4 access sequence index: An identifier to distinguish an access terminal before a MACID is assigned to it. This is used for initial access.

3.5 access terminal (AT): A device providing data connectivity to a user. An access terminal may be connected to a computing device such as a laptop personal computer or it may be a self-contained data device such as a personal digital assistant.

3.6 acquisition pilot: The first TDM pilot in the superframe preamble. This is used for initial acquisition.

3.7 active set: A set of sectors maintained in an access terminal that are assigned a MACID to the access terminal.

3.8 band class: A set of frequency channels and a numbering scheme for these channels.

3.9 base layer: *See: layered modulation.*

3.10 base station (BS): The device in the access network that communicates over the air interface, via one or more sectors, with the user terminals. Base stations coordinate the management of the air interface attributes.

3.11 beacon: A deterministic signal used to indicate the presence of an access network to the access terminals.

3.12 beacon orthogonal frequency division multiplexing (OFDM) symbol: A number of OFDM symbols allocated for transmission of the forward beacon pilot channel per every two superframes.

3.13 beacon subcarrier group: A subset of subcarriers belonging to the set of usable subcarriers in a beacon OFDM symbol.

3.14 block: A set of subcarriers.

3.15 block resource channel: A channel in which the hop-ports are mapped to a group of adjacent subcarriers in units of tiles.

3.16 CDMA hopping zone: The set of subcarriers over which the CDMA subsegments can be transmitted on the Reverse Link.

3.17 CDMA segment: The collection of all CDMA subsegments in the reverse link.

⁷The numbers in brackets correspond to those of the bibliography in Annex B.

3.18 CDMA subsegment: A collection of fixed number of subcarriers in reverse link allocated for the transmission of code division multiple access channels.

3.19 channel interleaver: The entity that permutes a sequence of symbols before transmission.

3.20 channel quality indicator: A quantized measurement of Forward Link channel quality.

3.21 Code Division Multiple Access (CDMA): A technique for spread-spectrum multiple-access digital communications that creates channels through the use of unique code sequences.

3.22 code symbol: The output of an error-correcting encoder. Information bits are input to the encoder and code symbols are output from the encoder. *See:* **block code**, **convolutional code**, **turbo code**, and **LDPC code**.

3.23 common pilot: A pilot in the forward common pilot channel.

3.24 common segment: A set of hop-ports present in all forward link PHY frames containing all the common forward control channels, where the common forward control channels consist of all channels carried on the forward link control segment except the forward shared control channel. The common segment may contain the forward shared control channel under certain conditions.

3.25 constituent code: A component of turbo code. One of the two recursive convolutional codes. *See:* **turbo code** and **convolutional code**.

3.26 convolutional code: A type of error-correcting code. A code symbol can be considered as the convolution of the input data sequence with the impulse response of a generator function.

3.27 constraint length: A parameter of a convolutional code. Constraint length is equal to one more than the number of registers in the convolutional code. *See:* **convolutional code**.

3.28 Coordinated Universal Time (UTC): An internationally agreed upon time scale maintained by the Bureau International des Poids et Mesures (BIPM) used as the time reference by nearly all commonly available time and frequency distribution systems (e.g., WWV, WWVH, LORAN-C, Transit, Omega, and GPS).

3.29 cyclic code: Cyclic code is a special type of linear block code such that a cyclic shift of a codeword is also a codeword.

3.30 cyclic prefix: In OFDM, to avoid inter-symbol-interference, a segment of time-domain signal at the end of the OFDM symbol is copied to the beginning, so that after the multipath channel, the time-domain signal is still cyclic within the sampling period.

3.31 cyclic redundancy check (CRC): A class of linear error detecting codes, which generate parity check bits by finding the remainder of a polynomial division. *See:* **cyclic redundancy check (CRC) bits**.

3.32 cyclic redundancy check (CRC) bits: The CRC check applied to named channels.

3.33 Dedicated Pilot: Pilot in Forward or Reverse Dedicated Pilot Channel.

3.34 DFT Matrix: A matrix representation of Discrete Fourier Transform.

3.35 DFT-Precoded CDMA: A time-domain CDMA sequence is broken up into 8 different subsequences of length 128. Each of these subsequences is converted to a frequency domain sequence through a Discrete Fourier Transform (DFT) operation, and is called DFT precoded CDMA.

3.36 Discrete Fourier Transform: A transform that converts a time domain sequence into a discrete frequency domain sequence.

3.37 Displaced Subcarrier: A concept introduced in the CDMA segment hopping in Reverse Link and Forward Link Control Segment hopping in the Forward Link. *See:* **Nominal CDMA Segment** and **Nominal FLCS Block**.

3.38 distributed resource channel: A channel in which the hop-ports are mapped to distributed subcarriers.

3.39 downlink (DL): The radio connection with the BS transmitting and the UT receiving. The term is used interchangeably with the term *forward link* or *FL*.

3.40 downloadable codebook: A set of precoding matrices that can be downloaded.

3.41 effective antenna: An effective antenna may be a single physical antenna or a linear combination of multiple physical antennas that appear to the receiver as a single physical antenna.

3.42 Effective Isotropically Radiated Power (EIRP): The product of the power supplied to the antenna and the antenna gain in a direction relative to an isotropic antenna.

3.43 Effective Radiated Power (ERP): The product of the power supplied to the antenna and its gain relative to a half-wave dipole in a given direction.

3.44 Encoder Tail Bits: A fixed sequence of bits added to the end of a block of data to reset the convolutional encoder to a known state.

3.45 enhancement layer: *See:* **Layered Modulation**.

3.46 Erasure Sequence: A sequence transmitted on a channel to hold it in the absence of data.

3.47 Exponential Sequence: A sequence of length L , whose i^{th} element, $E_{\omega}L(i)$, is given by $E_{\omega}^L = e^{-2\pi j\omega i/L}$ where $0 \leq i < L$, and j denotes the complex number $(0, 1)$.

3.48 Fast Fourier Transform (FFT): The Fast Fourier Transform is a discrete Fourier Transform algorithm that reduces the number of computations needed. *See:* **Discrete Fourier Transform**.

3.49 Forward Acknowledgment Channel: A portion of a Forward Channel used for the transmission of acknowledgments from an access network to multiple access terminals in response to the data received on the Reverse Code Division Multiple Access/Orthogonal Frequency Division Multiple Access Data Channel.

3.50 Forward Acquisition Channel: A channel sent on the Forward Link preamble consisting of one Orthogonal Frequency Division Multiplexing symbol to help in the initial acquisition process.

3.51 Forward Beacon Pilot Channel: The Forward Beacon Pilot Channel is used to indicate the presence of the access network to access terminals on other carriers.

3.52 Forward Broadcast and Multicast Pilot Channel: The Forward Broadcast and Multicast Pilot Channel is an unmodulated signal transmitted by an access network to provide a phase reference for coherent demodulation of the Forward Broadcast and Multicast Services Channel.

3.53 Forward Broadcast and Multicast Services Channel: The Forward Broadcast and Multicast Services Channel is a Forward Link channel carrying broadcast and multicast data.

3.54 Forward Cell Null Channel: The Forward Cell Null Channel defines subcarriers that are blanked by all the sectors in a cell. These subcarriers are used to measure the out-of-cell interference level.

3.55 Forward Channel Quality Indicator Pilot Channel: A signal transmitted by an access network to provide a reference for the measurement of the signals from the various antennas.

3.56 Forward Common Pilot Channel: An unmodulated signal transmitted by an access network to provides a phase reference for coherent demodulation and a means for signal strength comparisons between access networks for determining when to handoff.

3.57 Forward Data Channel: A portion of a Forward Link that carries higher-level data and control information from an access network to an access terminal.

3.58 Forward Dedicated Pilot Channel: An unmodulated signal transmitted by an access network that provides a phase reference for coherent demodulation of block resource channels (BRCHs).

3.59 Forward Error Correction: A process whereby data is encoded with block, convolutional, concatenated, LDPC, or turbo codes to assist in error correction of the link.

3.60 Forward Fast Other Sector Interference Channel: A channel sent on the Forward Link that carries an indication of other sector interference.

3.61 Forward Interference over Thermal Channel: A channel sent on the Forward Link that is used to indicate interference levels in a given Reverse Link hop-port subzone to access terminals in other sectors.

3.62 forward link (FL): The radio connection with the AN transmitting and the AT receiving. The term is used interchangeably with the term *downlink* or *DL*.

3.63 Forward Link Control Segment: A set of hop-ports or subcarriers allocated to transmit control messages to access terminals. This occurs in every PHY frame in which the Forward Acknowledgment Channel, the Forward Start-of-Packet Channel, the Forward Shared Control Channel, the Forward Fast Other Sector Interference Channel, the Forward Interference over Thermal Channel, the Forward Pilot Quality Indicator Channel, or the Forward Power Control Channel are multiplexed.

3.64 Forward Link Control Segment Available Subcarrier: A subcarrier that can be used for transmitting the Forward Link Control Segment.

3.65 Forward Link Control Segment Block: Each block in a set of hop-port blocks allocated to the Forward Link Control Segment.

3.66 Forward Link Control Segment Hopping Zone: The Forward Link Control Segment usable tiles are divided into three groups. Each group is a Forward Link Control Segment Hopping Zone. Each tile in the Forward Link Control Segment hops within one zone.

3.67 Forward Link Control Segment Resource: Each symbol in a Forward Link Control Segment is designed to have a diversity level of three, by being transmitted at three different subcarriers. This three-tuple is called an FLCS Resource.

3.68 Forward Link Control Segment Usable Tile: Tiles that can be used for transmitting the Forward Link Control Segment.

3.69 Forward Link Serving Sector: The sector that transmits the Forward Data Channel to the access terminal.

3.70 Forward Other Sector Interference Channel: A channel sent on the forward link preamble consisting of two Orthogonal Frequency Division Multiplexing symbols to help in the initial acquisition process. In addition, these symbols also carry the other sector interference value that is received from the SFP MAC protocol.

3.71 Forward Pilot Quality Indicator Channel: A channel sent on the forward link that indicates the strength of the reverse link for a given access terminal.

3.72 Forward Power Control Channel: A channel sent on the forward link that carries commands for closed-loop control of Reverse Link transmit power.

3.73 Forward Preamble Pilot Channel: A signal transmitted by an access network to aid in acquiring the system.

3.74 Forward Primary Broadcast Control Channel: A Forward Link Channel transmitted on the preamble that carries deployment-wide static parameters like cyclic prefix duration, number of guard carriers, in addition to the superframe index.

3.75 Forward Quick Paging Channel: A channel sent on the forward link preamble to aid the access terminal in identifying when a page is sent to it.

3.76 Forward Reverse Activity Bit Channel: A channel sent on the forward link that carries a single bit that indicates the load on the Reverse Link of a given access network.

3.77 Forward Secondary Broadcast Control Channel: A Forward Link Channel transmitted on the preamble that carries sufficient information to enable the access terminal to demodulate forward link data from the PHY frames, e.g., information on hopping patterns, pilot structure, control channel structures, effective antennas, multiplexing modes.

3.78 Forward Shared Control Channel: A channel sent on the forward link that carries control information for the Forward Data Channel transmission, as well as for group resource assignments.

3.79 Forward Start-of-Packet Channel: A channel sent on the forward link that is used to indicate to an access terminal whether a persistent assignment is still valid or if it has expired.

3.80 Forward Superposed Channel Quality Indicator Pilot Channel: A pilot channel that helps in the channel quality feedback for the superposed traffic on the broadcast and multicast section.

3.81 Forward Superposed Data Channel: A portion of a forward link that carries higher-level data and control information from an access network to an access terminal sent as superposed traffic on the broadcast and multicast section.

3.82 Forward Superposed Dedicated Pilot Channel: A pilot channel that helps in the channel estimation for the superposed traffic on the broadcast and multicast segment.

3.83 Fractional Frequency Reuse: A frequency reuse scheme where some of the subcarriers have a reuse factor less than one, i.e., they are not used in all sectors.

3.84 frame: A basic timing interval in the system, comprising of eight Orthogonal Frequency Division Multiplexing symbols.

3.85 Galois field (GF): A Galois field is a finite algebraic field with p_n elements where p is a prime number.

3.86 generator function: A function used to define a convolutional code. *See:* **convolutional code.**

- 3.87 generator polynomial:** A polynomial form of the generator function. *See:* **generator function.**
- 3.88 Global Asynchronous:** A system setup where the superframe timing of all sectors need not be aligned.
- 3.89 Global Hopping:** A hop-permutation used in the Reverse Link OFDMA Segment such that hop-ports are permuted across all subzones in the Global Hopping Zone. *See:* **hop-permutation, hop-port, subzone,** and **Reverse Link OFDMA Segment.**
- 3.90 Global Positioning System (GPS):** A U.S. government satellite system that provides location and time information to users.
- 3.91 Global Synchronous:** A system setup such that the superframes timing of all sectors are aligned within a predefined time interval.
- 3.92 GroupID:** A bit pattern assigned to a set of access terminals to identify them for the purpose of group resource assignment.
- 3.93 Group Resource Assignment:** The method of assigning shared resources to a set of access terminals on the Forward Data Channel.
- 3.94 Guard Interval:** A time interval, inserted between symbols to avoid intersymbol interference.
- 3.95 Guard Subcarrier:** A tone carrying no data to avoid intersymbol interference.
- 3.96 Hadamard matrix:** A square matrix formed by 1 and -1 s. All columns (rows) are orthogonal to each other. The columns (rows) of the matrix are Walsh Sequences. *See* Walsh Sequence.
- 3.97 HARQ:** A technique to transmit a packet by splitting it into several parts, which are transmitted sequentially through retransmissions. After each retransmission, the receiver attempts to decode the packet using all the received subpackets, the success of which may be fed back to the transmitter. The transmission can be early terminated if early decoding is successful.
- 3.98 HARQ Retransmission Index:** A counter for HARQ retransmissions.
- 3.99 hash function:** A function that outputs a unique number given an input number.
- 3.100 Hierarchical Modulation:** *See:* **Layered Modulation.**
- 3.101 hopping:** A pattern of frequency assignments.
- 3.102 hop-permutation:** A time-varying mapping from hop-port to subcarriers. *See:* **hop-port** and **subcarrier.**
- 3.103 hop-port:** An indexing scheme for subcarriers in an OFDM symbol.
- 3.104 hop-port index:** An index to address hop-ports.
- 3.105 hop-port subzone:** A collection of hop-ports of a fixed given size. *See:* **hop-port.**
- 3.106 hop-sequence:** A sequence of hop-permutations.
- 3.107 initial access:** Procedure for the access terminal to set up a connection to the access network for the first time, i.e., without a MACID.
- 3.108 Initial Protocol Set Identifier:** This is used to identify the default protocol stack for each of the major revisions of the air interface standard. The access network includes values of the supported Initial Protocol

Set Identifier in overhead messages. Multiple Protocol Set Identifiers may map to a single Initial Protocol Set Identifier.

3.109 interlace: A portioning of forward link or reverse link frames. HARQ transmissions use PHY frames on a given interlace.

3.110 interleaving: The process of permuting a sequence of symbols.

3.111 Inverse Fast Fourier Transform: An algorithm to efficiently compute the Inverse Fourier Transform.

3.112 knockdown codebook: A type of precoding matrix set used in Space Division Multiple Access (SDMA) constructed from a set of universal matrices. *See:* **universal matrix**.

3.113 layered modulation: A method to map multiple (two) bit streams to points in modulation constellation. Bits from the first bit stream (base layer) select one of the non-overlapping sets of constellation points. Bits from the second bit stream (enhancement layer) select one of the constellation points in the selected set. The constellation of the enhancement layer may be rotated for performance enhancement.

3.114 Linear Block Code: A type of block code such that the sum of two codewords is another codeword. *See* Block Code.

3.115 Link Assignment Block (LAB): A message for resource assignment.

3.116 Link Assignment Block (LAB) segment: A Forward Link Control Segment containing only the Forward Shared Control Channel.

3.117 local hopping: A hop-permutation used in the Reverse Link OFDMA Segment such that the hop-ports are permuted within a subzone. *See:* **hop-permutation**, **hop-port**, **subzone**, and **Reverse Link OFDMA Segment**.

3.118 logical antenna: A set of linear combination of physical antennas.

3.119 logical channel: Any channel carrying signaling and higher layer data. A number of logical channels are often multiplexed to form a physical channel.

3.120 Low-Density Parity Check Code: A type of error-correcting code. The code is defined as the null space of a sparse parity-check matrix.

3.121 MAC layer identifier: A bit pattern assigned to an access terminal by the MAC Layer for identification.

3.122 mean input power: The total received calorimetric power measured in a specified bandwidth at the antenna connector, including all internal and external signal and noise sources.

3.123 mean output power: The total transmitted calorimetric power measured in a specified bandwidth at the antenna connector when the transmitter is active.

3.124 modulation order: A measure for the size of a modulation constellation, and is defined as $\log_2(M)$, where M is the number constellation points in the modulation process.

3.125 Modulation Symbol: For the Reverse CDMA Data Channel, the Reverse OFDMA Data Channel, the Forward Data Channel, the Forward Broadcast and Multicast Services Channel and the Forward Superposed

Channel, a modulation symbol is defined as the output of the QPSK/8-PSK/16-QAM/64-QAM modulator. For all other channels, a modulation symbol is defined as the input to the signal point mapping block and the output of the interleaver or the sequence repetition block, if present.

3.126 Modulo-2 Addition: An operation of two binary numbers with the output being the exclusive-or of the two inputs.

3.127 Multiple Codeword: A Multiple Input Multiple Output transmission mode where multiple codes are used to encode the packet being transmitted over the various antennas.

3.128 Multiple Input Multiple Output: An abstract mathematical model for multi-antenna communication systems where the transmitter has multiple antennas capable of transmitting independent signals and the receiver is equipped with multiple receive antennas.

3.129 mutually synchronous: A set of access networks whose timing is synchronized within a pre-specified limit.

3.130 Newly-Freed Subcarrier: A concept introduced in the CDMA segment hopping in Reverse Link and Forward Link Control Segment hopping in the forward link. *See:* **Nominal CDMA Subsegment** and **Nominal FLCS Block**.

3.131 Nominal CDMA subsegment: In the reverse link, the CDMA subsegments are designed to hop across the all available subcarriers. First, a set of subcarriers is assumed for the CDMA subsegments, which is defined as Nominal CDMA subsegments. The other hop-ports are mapped to available subcarriers not occupied by the Nominal CDMA subsegments. Then, the CDMA subsegments can be hopped to other locations. The subcarriers that the CDMA subsegments hop to are defined as Displaced Subcarriers, and the subcarriers that the CDMA subsegments leave are defined as Newly-Freed Subcarriers. The other hop-ports that originally map to the Displaced Subcarriers are adjusted to map to the Newly-Freed Subcarriers.

3.132 Nominal FLCS Block: In the forward link, the FLCS Blocks are designed to hop across the all available subcarriers. First, a set of subcarriers is assumed for the FLCS Blocks, which is defined as Nominal FLCS Blocks. The other hop-ports are mapped to available subcarriers not occupied by the Nominal FLCS Blocks. Then, the FLCS Blocks can be hopped to other locations. The subcarriers that the FLCS Blocks hop to are defined as Displaced Subcarriers, and the subcarriers that the FLCS Blocks leave are defined as Newly-Freed Subcarriers. The other hop-ports that originally map to the Displaced Subcarriers are adjusted to map to the Newly-Freed Subcarriers.

3.133 Orthogonal Frequency Division Multiple Access: A multi-user version of the Orthogonal Frequency Division Multiplexing digital modulation scheme. Multiple access is achieved in Orthogonal Frequency Division Multiple Access by assigning subsets of subcarriers to individual users.

3.134 Orthogonal Frequency Division Multiple Access (OFDMA) segment: The set of subcarriers used to transmit channels using OFDMA.

3.135 Orthogonal Frequency-Division Multiplexing (OFDM): A modulation technique that utilizes multiplexing based on orthogonal complex harmonic basis functions together with a cyclic prefix to allow multi-path resilience.

3.136 Orthogonal Frequency-Division Multiplexing (OFDM) Symbol: An Orthogonal Frequency Division Multiplexing symbol is comprised of individually modulated subcarriers that carry complex-valued data.

3.137 packet: A sequence of information bits to be transmitted by the Physical Layer.

3.138 Packet Error Detection: A method used to detect whether the received packet is in error. Usually, the data is encoded with a Cyclic Redundancy Check to aid in this detection. *See:* **Cyclic Redundancy Check**.

3.139 Packet Format: A specification of spectral efficiency, packet size, code rate and modulation order to be used for each transmission.

3.140 Packet Format Index: The index of Packet Format.

3.141 parity bits: Redundant bits added in an error-correcting code that help in detecting if a packet was demodulated properly.

3.142 parity symbol: Redundant symbols added in an error-correcting code that help in detecting if a packet was demodulated properly.

3.143 permutation function: A function that permutes a sequence of input bits, symbols, etc.

3.144 permutation matrix: A matrix form of the permutation function.

3.145 persistent assignment: An assignment that is valid until explicitly deassigned.

3.146 physical antenna: A radiating radio element.

3.147 Physical Layer: The part of the communication protocol between the access terminal and the access network that is responsible for the transmission and reception of data. The Physical Layer in the transmitting station is presented a frame and transforms it into an over-the-air waveform. The Physical Layer in the receiving station transforms the waveform back into a frame.

3.148 PilotID: A 10-bit quantity derived from the PilotPN and the IsSynchronous field of the Overhead Messages Protocol. PilotID is useful in distinguishing between two sectors with the same PilotPN provided that they have different values of the IsSynchronous field.

3.149 PilotPhase: A 9-bit quantity defined for use in modulating the Forward Other Sector Interference Channel. The PilotPhase of an access network depends on the superframe index and the PilotPN.

3.150 PilotPN: A 9-bit identifier used in the Physical Layer for differentiating between different sectors. Sectors with different PilotPNs transmit different acquisition pilots on the Forward Other Sector Interference Channel, which enables access terminals to monitor the signal strengths of the two sectors. Similarly, two sectors with different PilotPNs use different pseudo-random scrambling sequences on the Forward Data Channel.

3.151 power amplifier: An analog device that amplifies the radio frequency signal before sending it through the antenna.

3.152 Power density: The expected energy per modulation symbol, where “modulation symbol” refers to frequency domain (pre-IFFT) symbols, as defined in the Physical Layer specification. The “expected” power is used to cover the case of transmissions using non-constant modulus signal constellations. In all cases, the density shall be computed according to the parameters of the modulation symbols in the particular channel being specified.

NOTE—Note that this definition of power density is not true in the strictest sense because of cyclic prefix and windowing overheads. However, it provides a convenient method of communicating power levels to the Physical Layer Protocol. The construction of transmitted waveforms as a function of power densities is specified by the Physical Layer Protocol.⁸

3.153 preamble: *See: Superframe Preamble.*

⁸Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

- 3.154 precoder index:** The index to a precoding matrix in a set of precoding matrices.
- 3.155 precoding:** A method to beamform with multiple antennas to focus a spatial beam in a certain direction.
- 3.156 precoding matrix:** A matrix, whose columns determine the transmit vectors used for precoding during the Space Division Multiple Access operation.
- 3.157 Precoding Matrix Cluster:** A subset of precoding matrix set that cover the whole space. Different clusters of precoding matrices are used for the Space Division Multiple Access operation. *See:* **precoding matrix**.
- 3.158 preferred matrix:** A precoding matrix that the access terminal chooses for the future SDMA transmissions. *See:* **Precoding Matrix**.
- 3.159 Preferred Matrix Index:** The index to the preferred matrix used by access terminal to request the preferred matrix. *See:* **preferred matrix**.
- 3.160 Protocol Set Identifier:** Used to identify the subtype of each of the protocols in the protocol stack.
- 3.161 Pruned Bit Reversal Interleaver:** An interleaver such that the output ordering is the reverse of the binary representations of all input locations.
- 3.162 Pseudonoise (PN) Register:** A shift register structure used in the Pseudonoise (PN) sequence generator.
- 3.163 Pseudonoise (PN) sequence:** A periodic binary sequence.
- 3.164 Punctured Code:** An error-correcting code generated from another error-correcting code by deleting (i.e., puncturing) code symbols from the encoder output.
- 3.165 Radio Configuration:** A set of Broadcast and Multicast Serviced Channel transmission formats that are characterized by Physical Layer parameters such as Orthogonal Frequency Division Multiplexing symbol duration.
- 3.166 rank:** The number of effective antennas used in an MIMO transmission.
- 3.167 Rate-1/3 Concatenated Code:** A linear block code with 4 input bits and 12 output bits that can be interpreted as a concatenation of a rate-2/3 cyclic code and a rate-1/2 tail-biting convolutional code.
- 3.168 readymade codebook:** A type of precoding matrix set.
- 3.169 Recursive Convolutional Code:** A convolutional code that uses feedback.
- 3.170 Reed-Solomon Code:** A type of linear block code.
- 3.171 Reference Receive Power Density:** The reference receive power density used by the access terminal as the reference power density per subcarrier for comparing the strengths of different sectors, and is defined as

$$\frac{P_{\text{PBCH}}}{n_{\text{sc}}},$$

where

P_{PBCH} denotes the total received power of Forward Primary Broadcast Control Channel,

n_{sc} is the number of subcarriers carrying the Forward Primary Broadcast Control Channel in each OFDM symbol.

3.172 Reference Receive Power Level: The received power of the Forward Primary Broadcast Control Channel is used by the access terminal as a Reference Received Power Level for the transmitting sector.

3.173 Reference Transmit Power: The power at which the first OFDM symbol of the preamble (Forward Primary Broadcast Control Channel) is transmitted, and may vary from sector to sector.

3.174 reserved subzone: A forward link subzone that is reserved for non-unicast transmissions. A portion of this can be used for broadcast and multicast services. This subzone cannot be assigned to block resource channel and distributed resource channel zones.

3.175 Resource Multiplexing Mode: Methods to allocate forward link hop-ports to block resource channel and distributed resource channel zones.

3.176 ReuseIndex: $\text{PilotPhase} \bmod 8$. *See:* **PilotPhase**.

3.177 Reverse Access Channel: A Reverse Code Division Multiple Access Channel used by the access terminal for communicating with the access networks. The Access Channel is used for short signaling message exchanges, such as call originations, responses to pages, and registrations. The Access Channel is a slotted random access channel.

3.178 Reverse Acknowledgment Channel: A portion of a Reverse Orthogonal Frequency Division Multiple Access Channel used for the transmission of acknowledgments from an access terminal to multiple access networks in response to the data received on the Forward Packet Data Channel.

3.179 Reverse Acknowledgment Channel Partial-Tile: A contiguous set of eight subcarriers for a duration of two, four, or eight OFDM symbols that is used for the Reverse Acknowledgment Channel. The partial-tile is further divided into one, two, or four subtiles, each occupying a eight subcarriers by two OFDM symbols block.

3.180 Reverse Acknowledgment Channel Subtile: *See:* **Reverse Acknowledgment Channel Partial-Tile**.

3.181 Reverse Auxiliary Pilot Channel: An unmodulated signal transmitted in the Reverse Link CDMA segment by an access terminal in conjunction with the Reverse CDMA Data Channel. This channel provides a phase reference for coherent demodulation for demodulation of the Reverse CDMA Data Channel.

3.182 Reverse Code Division Multiple Access Dedicated Control Channel: A portion of a Code Division Multiple Access Reverse Link that carries control information and other feedback information from an access terminal to an access network.

3.183 Reverse Code Division Multiple Access Data Channel: A portion of a Code Division Multiple Access Reverse Link that carries higher-level data and control information from an access terminal to an access network.

3.184 Reverse Control Channel: A portion of a Reverse Link that carries control information from an access terminal to an access network.

3.185 Reverse Dedicated Pilot Channel: An unmodulated signal transmitted in the Reverse Link Orthogonal Frequency Division Multiple Access segment by an access terminal. This channel provides a phase reference for coherent demodulation of the Reverse Orthogonal Frequency Division Multiple Access Traffic Channels.

3.186 Reverse Code Division Multiple Access Control Channel: A portion of an Orthogonal Frequency Division Multiple Access Reverse Link that carries control information and other feedback information from an access terminal to an access network.

3.187 reverse link (RL): The radio connection with the AT transmitting and the AN receiving. The term is used interchangeably with the term *uplink* or *UL*.

3.188 Reverse Link Serving Sector: A sector to which the access terminal is scheduled to transmit its Reverse Link traffic data.

3.189 Reverse OFDMA Dedicated Control Channel Quarter-Tile: A resource spanning eight contiguous subcarriers in frequency and four contiguous OFDM symbols in time.

3.190 Reverse Orthogonal Frequency Division Multiple Access Control Channel: A portion of a OFDMA Reverse Link that carries control information and other feedback information from an access terminal to an access network.

3.191 Reverse Orthogonal Frequency Division Multiple Access Data Channel: A portion of a Orthogonal Frequency Division Multiple Access Reverse Link that carries higher-level data and control information from an access terminal to an access network.

3.192 Reverse Pilot Channel: An unmodulated signal transmitted in the Reverse Link Code Division Multiple Access segment by an access terminal. A reverse pilot channel may provide a phase reference for coherent demodulation and a means for signal strength measurement.

3.193 Reverse Traffic Channel: One or more code channels used to transport user and signaling traffic from the access terminal to the access network. *See:* **Reverse Code Division Multiple Access Data Channel** and **Reverse Orthogonal Frequency Division Multiple Access Data Channel**.

3.194 Rotational OFDM: A modulation technique in which a set of subcarriers are transformed through a rotational matrix for robustness in frequency selective fading channels.

3.195 route: Each InUse protocol stack instance is called a route.

3.196 Scrambling Algorithm: An algorithm that randomizes an input sequence using a pseudonoise sequence. The pseudonoise sequence is generated using a shift register structure whose initial state is given by a seed. *See:* **Seed**, **Shift Register**, and **Pseudonoise (PN) sequence**.

3.197 SDMA Subtree: A portion of the hop-ports used for SDMA transmissions. Typically, the channel tree contains multiple subtrees such that there is one subtree per SDMA cluster with one primary subtree. Identical hop patterns are used across the subtrees.

3.198 sector: One set of physical layer channels transmitted between base station and the user terminals within a given frequency assignment.

3.199 SectorSeed: An integer number used to initialize the state of the Pseudonoise generator that is sector specific.

3.200 Seed: An integer number used to initialize the state of the Pseudonoise generator.

3.201 SFNCellID: SFNCellID is a nine-bit quantity and is a field of the Overhead Messages Protocol. *See:* **SFNPhase**.

3.202 SFNID: A bit pattern assigned to a set of access networks that transmits data in a Single Frequency Network mode.

3.203 SFNPhase: In Synchronous mode, SFNPhase depends on the SuperframeIndex and is equal to $(\text{SFNCellID} + \text{SuperframeIndex}) \bmod 512$. In Asynchronous mode, SFNPhase is equal to SFNCellID.

3.204 Shared Control Channel Usable Hop-Port: A set of hop-ports that can be used for the transmission of the Forward Shared Control Channel.

3.205 Shift Register: A circuit formed by sequentially connected registers and feedback.

3.206 Silence Interval: A set of eight consecutive OFDM symbols (or multiples thereof) on the reverse link where no signal is transmitted by the access terminal.

3.207 Single Codeword: A Multiple Input Multiple Output transmission mode where a single code is used to encode the packet being transmitted over the various antennas.

3.208 Single Frequency Network: A set of sectors synchronously transmitting the same waveform on the same frequency assignment, with the exception of a sector-dependent delay and gain, within a specified frequency-division and time-division multiplexed channel.

3.209 Single Input Single Output: An abstract mathematical model for single-antenna communication systems where the transmitter has a single antennas capable of transmitting signals and the receiver is equipped with a single receive antenna.

3.210 Space Division Multiple Access: This technique spatially separates and multiplexes users through adaptive beamforming, thereby providing gains in system throughput by reducing interference across users.

3.211 Space Time Transmit Diversity: A technique to transmit multiple transformed versions of a data stream across a number of antennas and to exploit the various received versions of the data to improve the reliability of data-transfer.

3.212 Spatial Beam: A precoded transmission from a set of antennas used in Space Division Multiple Access, which typically creates a spatial pattern causing the transmission to be localized in a certain space.

3.213 subcarrier: Each frequency of the Discrete Fourier Transform.

3.214 subcarrier-symbol: One hop-port for the duration of one OFDM symbol. This is the smallest unit in the time-frequency grid and is the smallest unit that can be utilized to populate a modulation symbol.

3.215 subpacket: A subdivision of data packets whose size is subject to a pre-defined upper limit. Each subpacket may be independently encoded.

3.216 subzone: A set of subcarriers or hop-ports of given size, usually 64 or 128.

3.217 Supercast: A technique to overlay unicast data stream over the BCMCS data stream.

3.218 Superframe: A structure formed by a Superframe Preamble and multiple PHY frames. *See: Superframe Preamble and frame.*

3.219 Superframe Index: The index of Superframe.

3.220 Superframe Preamble: A collection of eight OFDM symbols transmitted at the beginning of each forward link superframe. *See: Superframe.*

3.221 System Time: The time reference used by the system. System Time is synchronous to UTC time (except for leap seconds) and uses the same time origin as GPS time. All access networks use the same System Time (within a small error tolerance). Access terminals use the same System Time, offset by the propagation delay from the access network to the access terminal. *See: Coordinated Universal Time.*

3.222 tail bit: *See: Encoder Tail Bit.*

3.223 Tail-biting Convolutional Code: A modified type of convolutional code, where the initial state of the trellis is selected such that the final state of the trellis is the same as the initial state. No tail bits are added.

3.224 TDM Pilot: The last three OFDM symbols in the Superframe Preamble. *See: Superframe Preamble.*

3.225 tile: A group of 16 hop-ports for the duration of 8 OFDM symbols. In the case of the Forward Superposed Channel, a tile may contain a different number of OFDM symbols as dictated by the Radio Configuration of supercast.

3.226 tile antenna: A linear combination of physical antennas, where the choice of the linear combination is fixed over each tile, but may vary from tile to tile.

3.227 tile segment: A portion of a tile used for defining FLCS resources. *See: tile.*

3.228 time reference: A reference established by the access terminal that is synchronous with the earliest arriving multipath component used for demodulation.

3.229 Turbo Code: A type of error-correcting code. A code symbol is based on the outputs of the two recursive convolutional codes (constituent codes) of the Turbo Code.

3.230 Turbo Interleaver: A component of Turbo Code. The information sequence is interleaved using the Turbo Interleaver before being input to the second constituent convolutional code. *See: Turbo Code and Constituent Convolutional Code.*

3.231 Universal Matrix: One or two matrices from which a Knockdown Codebook is constructed. Examples include the Identity matrix and the Fourier matrix.

3.232 uplink (UL): The radio connection with the user terminal transmitting and the base station receiving. The term is used interchangeably with the term *reverse link* or *RL*.

3.233 user terminal (UT): A device providing data connectivity to an end user device (EUD) user. A user terminal may be connected to a computing device, such as a laptop personal computer, or it may be a self-contained data device, such as a personal digital assistant.

3.234 vector channel quality indicator (VCQI): VCQI is used to convey extra channel quality information to the access network, beyond the information that is carried in the CQI channel defined by the Physical Layer.

3.235 Walsh chip: The shortest identifiable component of a Walsh function. There are 2^N Walsh chips in one Walsh function where N is the order of the Walsh function.

3.236 Walsh function: One of 2^N time orthogonal binary functions (note that the functions are orthogonal after mapping '0' to 1 and '1' to -1). The n^{th} Walsh function ($n = 0$ to $N - 1$) after the mapping to ± 1 symbols is denoted by $W_n N$.

3.237 Walsh sequence: *See: Walsh function.*

3.238 windowing function: A time domain function used to shape the amplitude of signal in the guard interval. *See:* **guard interval**.

3.239 zone: Either Distributed Resource Channel Zone or Block Resource Channel Zone in the forward link.

4. Abbreviations and acronyms

16-QAM	16-ary quadrature amplitude modulation
64-QAM	64-ary quadrature amplitude modulation
8-PSK	8-ary phase shift keying
AF	assured forwarding
AN	access node
ASMP	Active Set Management Protocol
AT	access terminal
ATI	access terminal identifier
BCMCS	Broadcast and Multicast Services
BE	best effort
BPSK	binary phase shift keying
BRCH	block resource channel
BSP	Basic Signaling Protocol
CDMA	Code Division Multiple Access
CQI	channel quality indicator
CRC	cyclic redundancy check
DFT	Discrete Fourier Transform
DL	downlink
DLC	Data Link Control
DRCH	distributed resource channel
DSCP	Differential Service Code Point
EF	expedited forwarding
EIRP	Effective Isotropically Radiated Power
ERP	Effective Radiated Power
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform
FL	forward link
FLCS	Forward Link Control Segment
FLSS	Forward Link Serving Sector
GCL Sequence	Generalized Chirp Like Sequence
GF	Galois field
GH	global hopping
GPS	Global Positioning System
GRA	group resource assignment
IFFT	Inverse Fast Fourier Transform
IPSI	Initial Protocol Set Identifier
LAB	Link Assignment Block
LDPC Code	Low-Density Parity Check Code
LH	local hopping
LLC	Logical Link Control
LSB	least significant bit

MAC Layer	Medium Access Control Layer
MACID	MAC Layer Identifier
MCW	multiple codeword
MIMO	Multiple Input Multiple Output
MSB	most significant bit
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OMP	Overhead Messages Protocol
PBRI	Pruned Bit Reversal Interleaver
PCI	Precoder Index
PDU	Protocol Data Unit
PF	packet format
PHY	Physical Layer
PN	Pseudonoise
PSI	Protocol Set Identifier
QPSK	quadrature phase shift keying
RC	radio configuration
RCC	Reverse Control Channel
RL	reverse link
RLC	Radio Link Control
RLP	Radio Link Protocol
RLSS	Reverse Link Serving Sector
RMU	RLC Message Unit
SAP	Service Access Point
SCW	single codeword
SDMA	Space Division Multiple Access
SE	Spectral Efficiency
SFN	Single Frequency Network
SISO	Single Input Single Output
SMB	Short Message Broadcast
SPF	Superframe Preamble
STTD	Space Time Transmit Diversity
TDD	Time Division Duplex
UL	uplink
UT	user terminal
UTC	Coordinated Universal Time or Temps Universel Coordonné <i>See</i> Coordinated Universal Time
VCQI	vector channel quality indicator
VoIP	Voice over Internet Protocol

5. General introduction

5.1 General

These technical requirements form a compatibility standard for mobile broadband wireless access systems. The requirements ensure that a compliant access terminal (AT) or user terminal (UT) can obtain service through any access network or base station (BS) conforming to properly selected modes of this standard, consistent with equipment and operator requirements, thus providing a framework for the rapid development of cost-effective, interoperable multivendor mobile broadband wireless access systems. This compatibility standard is targeted for use in a wide variety of licensed frequency bands and regulatory environments.

This standard omits details concerning a particular access network implementation. However, it is designed to allow both a fixed hierarchical backhaul structure or a more dynamic and non-hierarchical backhaul structure. The term *access network* itself is used throughout the document to remind the reader that both implementations are possible, and to avoid prejudicing one implementation over another. This standard is primarily oriented toward requirements necessary for the design and implementation of the access/user terminals. Thus, more detailed procedures are specified for access/user terminal operation in an effort to ensure a uniform response to all access networks.

This standard includes provisions for future service additions and expansion of system capabilities. The architecture defined by this standard permits such expansion without the loss of backward compatibility.

5.2 Modes of the standard

This standard specifies two modes of operation, a Wideband mode and a 625k-MC mode, both of which have their own distinct and optimized MAC and PHY layers. The Wideband mode is designed to operate for all Frequency division Duplex (FDD) and Time Division Duplex (TDD) bandwidths and is described in 5.4, Clause 2 through Clause 17, and Annex A. The 625k-MC mode is designed with 625 kHz carrier bandwidth supporting aggregation of multiple carriers for TDD operation only and is described in 5.5, Clause 18 through Clause 31, and Annex A.

This standard does not require any device to support multiple modes, and interoperability between terminals and access networks is only assured with each mode. This standard is designed to ensure multivendor interoperability of terminals and network equipment within each mode if that equipment fully complies with the procedures, message sequences, and protocols specified in this standard for each mode.

5.3 Requirements language

The words “shall” and “shall not” identify requirements to be followed strictly to conform to the standard and from which no deviation is permitted. The words “should” and “should not” indicate that one of several possibilities is recommended as particularly suitable, without mentioning or excluding others, that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. The words “may” and “need not” indicate a course of action permissible within the limits of the standard. The words “can” and “cannot” are used for statements of possibility and capability, whether material, physical, or causal.

5.4 Wideband mode overview

For the Wideband mode, the specifications in this standard apply to both FDD and TDD, unless otherwise stated.

5.4.1 Architecture reference model

The Wideband architecture reference model is presented in Figure 1.

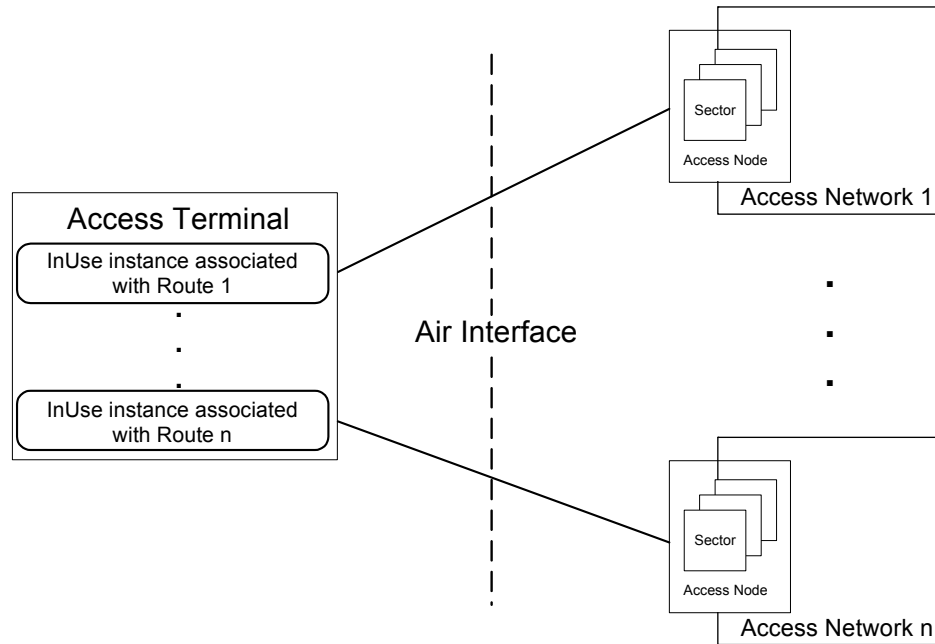


Figure 1—Wideband architecture reference model

The functional units of the reference architecture in Figure 1 are defined in the definition clause and included here for convenience of the reader:

- *access node (AN)*: The physical device that implements the MAC and PHY of one or more sectors in the network.
- *access network*: Access network is a collection of ANs.
- *access terminal (AT)*: A device providing data connectivity to a user. An access terminal may be connected to a computing device such as a laptop personal computer or it may be a self-contained data device such as a personal digital assistant.
- *forward link (FL)*: The radio connection with the AN transmitting and the AT receiving. The term is used interchangeably with the term *downlink* or *DL*.
- *reverse link (RL)*: The radio connection with the AT transmitting and the AN receiving. The term is used interchangeably with the term *uplink* or *UL*.
- *route*: Each InUse protocol stack instance is called a route.
- *sector*: One set of physical layer channels transmitted between the access network and the access terminals within a given frequency assignment.

The definition of InUse and InConfiguration Protocols is found in 5.4.9.

The reference model includes the air interface between the access terminal and the access node. The access terminal communicates with one or more access networks over the air interface. An access network may not support a radio link. An access network that does not have direct radio communication with the access terminal (either because the access network does not support a radio link or because the access network is not the current serving access network) can communicate with the access terminal by sending/receiving Route Protocol packets through the serving access network using the Inter-Route Tunneling Protocol of the serving access network.

The access terminal maintains an InUse instance of the protocol stack associated with each access network with which it is communicating. Each InUse protocol stack instance is called a route. The protocols used over the air interface are defined in this standard. This standard describes the interaction between one InUse instance at the access terminal and a corresponding InUse instance at one access network. Unless specified otherwise, access terminal and access network procedures are applicable only to the InUse instance being referenced. For example,

- “The access terminal shall transition to Inactive State” shall be interpreted to mean that “This InUse instance of the access terminal shall transition to the Inactive State”.

When a reference to more than one InUse instance is required, the reference is explicitly stated. For example,

- “If ExampleVariable is set to zero in all InUse instances at the access terminal, the access terminal shall issue an *Example* command,” shall be interpreted to mean that the instance of the protocol currently being referred to shall check ExampleVariable across all InUse instances, and issue the *Example* command if the condition is satisfied.

In procedural descriptions throughout the text describing operations in the Wideband mode, the AT and the AN shall perform the indicated steps in the sequence given, unless otherwise stated.

5.4.2 Protocol architecture

The air interface has been layered, with interfaces defined for each layer (and for each protocol within each layer). This allows future modifications to a layer or to a protocol to be isolated.

5.4.2.1 Layers

Figure 2 describes the layering architecture for each route of the air interface that carries unicast data. Each sublayer or plane consists of one or more protocols that perform the sublayer’s functionality.

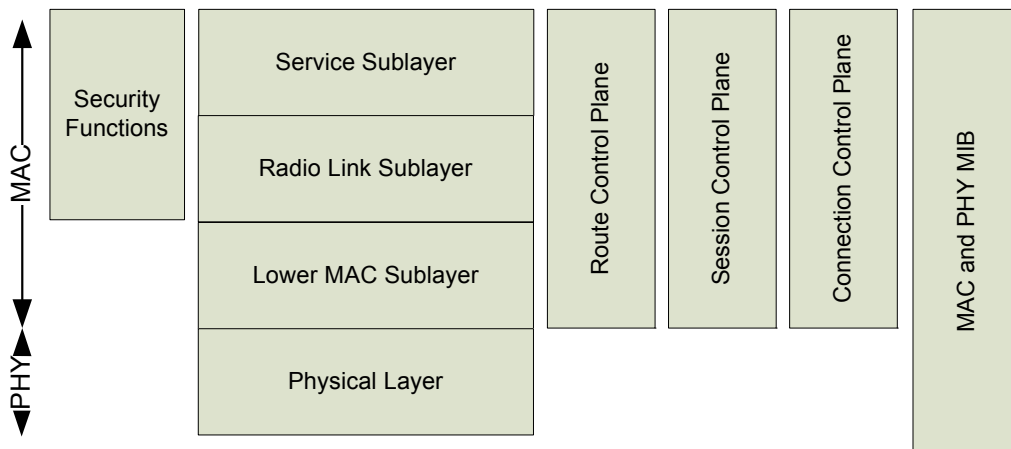


Figure 2—Unicast Route Layering architecture

The layers specified in Figure 2 are as follows:

- *Physical Layer*: The Physical Layer provides the channel structure, frequency, power output, modulation, and encoding specifications for the Forward and Reverse Channels. The Physical Layer protocol is defined in Clause 9.

- *Lower MAC Sublayer*: The Lower Medium Access Control (MAC) Sublayer defines the procedures used to receive and to transmit over the Physical Layer. The Lower MAC Sublayer is defined in Clause 8.
- *Radio Link Sublayer*: Protocols in the Radio Link Sublayer provide services such as reliable and in-sequence delivery of Application Sublayer packets, multiplexing of Application Sublayer packets, and Quality of Service (QoS) negotiation in support of applications. The Radio Link Sublayer is defined in Clause 7.
- *Service Sublayer*: The Service Sublayer provides support for multiple applications. It provides the Signaling Protocol for transporting air interface protocol messages. It also provides the Inter-Route Tunneling Protocol for transporting packets to/from other routes. Other examples of Service Sublayer protocols include the Extensible Authentication Protocol (EAP) Support Protocol for support of authentication, the Robust Header Compression (RoHC) Support Protocol for compressing the packet header, and protocols for transporting packets from other air interfaces. These Service Sublayer protocols are defined in Clause 6.
- *Connection Control Plane*: The Connection Control Plane provides air link connection establishment and maintenance services. The Connection Control Plane is defined in Clause 11.
- *Session Control Plane*: The Session Control Plane provides protocol negotiation and protocol configuration services. The Session Control Plane is defined in Clause 12.
- *Route Control Plane*: The Route Control Plane provides creation, maintenance, and deletion of routes. The Route Control Plane is defined in Clause 13.
- *Security Functions*: Security functions include functions for key exchange, ciphering, and message integrity protection. Security functions are defined in Clause 10.
- *MAC and PHY MIB*: For unicast transport, the MAC and PHY Management Information Base (MIB) provides usage statistics gathered from all protocols and management planes, e.g., access attempts, bytes successfully transferred, dropped connections. The MAC and PHY MIB is defined in Clause 17.

Figure 3 describes the layering architecture for a route of the air interface that carries broadcast-multicast service (BCMCS) data. Each layer or plane consists of one or more protocols that perform the layer's functionality.

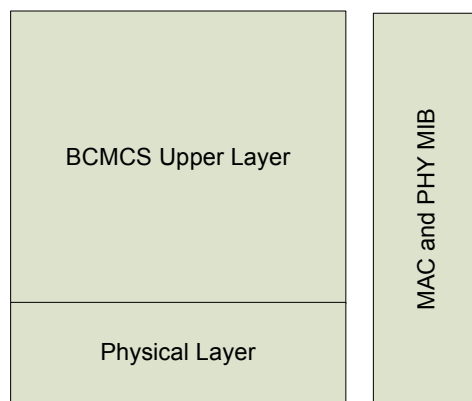


Figure 3—BCMCS Route Layering architecture

- *Physical Layer*: The Physical Layer provides the channel structure, frequency, power output, modulation, and encoding specifications for the BCMCS channel. The Physical Layer protocol is defined in Clause 9.
- *Broadcast-Multicast Service Upper Layer*: Besides the unicast routes, the air interface also defines a BCMCS Route that contains protocols in support of Broadcast-Multicast Service (BCMCS). BCMCS Upper Layer protocols are defined in Clause 14.

Each layer contains one or more protocols. Protocols use signaling messages or headers to convey state information to their peer protocols at the other side of the air-link. When protocols and applications send messages, they use the Signaling Protocol over the Radio Link Protocol to transmit these messages.

5.4.2.2 Protocols

Each sublayer specifies one or more protocol types. Protocols are associated with a type that denotes the type of the protocol (e.g., Access Channel MAC Protocol) and with a subtype that denotes a specific instance of a protocol (e.g., the Basic Access Channel MAC Protocol).

The following subclauses provide a brief description of protocols in each layer. A more complete description is provided in the introduction subclause of each layer.

5.4.2.2.1 Services Sublayer

The protocols of the Service Sublayer are illustrated in Figure 4 and described in the text following.

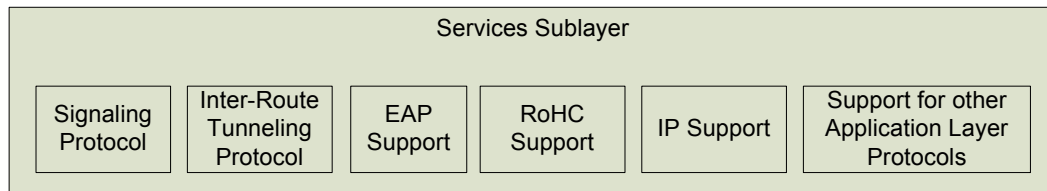


Figure 4—Services Sublayer Protocols

- *Signaling Protocol*: The Signaling Protocol provides message transmission services for signaling messages.
- *Inter-Route Tunneling Protocol*: The Inter-Route Tunneling Protocol provides transport of packets from other routes.
- Support for other Application Sublayer Protocols, such as EAP Support Protocol, RoHC Support Protocol etc., may create payload to be carried over this air interface. Examples of other protocols include IEEE 802.1Q C-VLAN frames IEEE 802.1Q and IEEE 802.3 Ethernet PDUs (see IEEE Std 802.3⁹).

⁹Information of references can be found in Clause 2.

5.4.2.2.2 Radio Link Sublayer

The protocols of the Radio Link Sublayer are illustrated in Figure 5 and described in the text following.

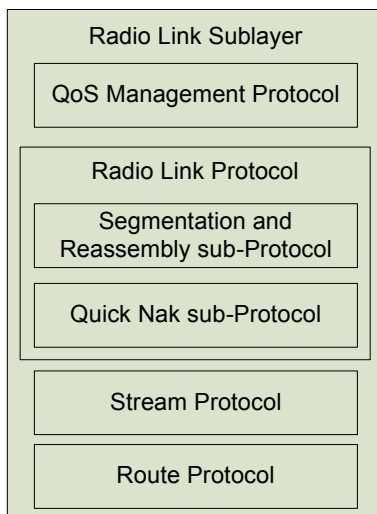


Figure 5—Radio Link Sublayer protocols

The Radio Link Sublayer defines the following four protocols:

- *Radio Link Protocol*: The Radio Link Protocol (RLP) is a bearer protocol that provides segmentation and re-assembly, ACK/NAK, retransmission, and duplicate detection for upper layer packets. The goal is to improve a voice-quality channel to provide packet-quality reliability.
- *Stream Protocol*: The Stream Protocol is a bearer protocol that identifies the stream on which the upper layer fragments are being carried.
- *Route Protocol*: The Route Protocol is a bearer protocol that identifies and routes Stream Protocol packets between the access terminal and the access network.
- *QoS Management Protocol*: The QoS Management Protocol provides negotiation of flow and filter specifications to provide appropriate Quality of Service (QoS) to application sublayer packets.

5.4.2.2.3 Lower MAC Sublayer

The protocols of the Lower MAC Sublayer are illustrated in Figure 6 and described in the text following.

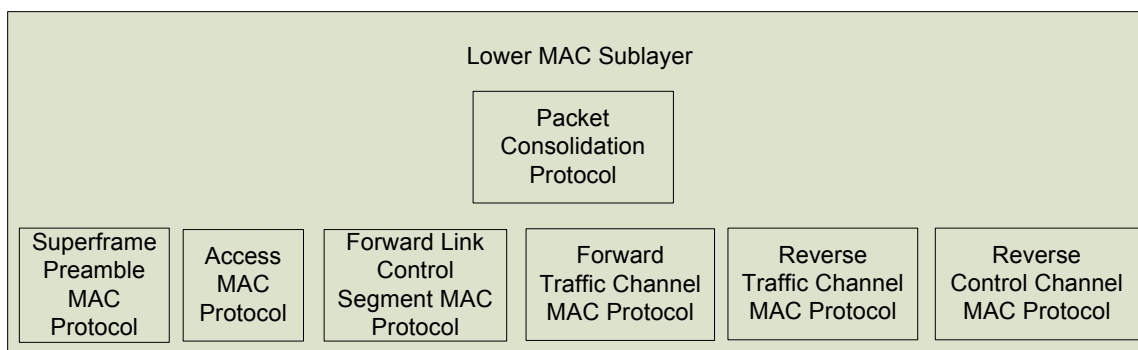


Figure 6—Lower MAC Sublayer Protocols

The Lower MAC Sublayer contains the following seven protocols:

- *Packet Consolidation Protocol*: Provides transmit prioritization and packet encapsulation for upper layer packets.
- *Superframe Preamble MAC Protocol*: Provides procedures followed by the access network to transmit and access terminal to receive the superframe preamble.
- *Access Channel MAC Protocol*: Provides the procedures followed by the access terminal to transmit, and by the access network to receive the Access Channel.
- *Forward Link Control Segment MAC Protocol*: Provides the procedures followed by the access network to transmit, and by the access terminal to receive the Forward Control Channel.
- *Forward Traffic Channel MAC Protocol*: Provides the procedures followed by the access network to transmit, and by the access terminal to receive the Forward Data Channel.
- *Reverse Control Channel MAC Protocol*: Provides the procedures followed by the access terminal to transmit, and by the access network to receive the Reverse Control Channel.
- *Reverse Traffic Channel MAC Protocol*: Provides the procedures followed by the access terminal to transmit, and by the access network to receive the Reverse Data Channel.

5.4.2.2.4 Physical Layer

The Physical Layer defines the Physical Layer Protocol as shown in Figure 7.

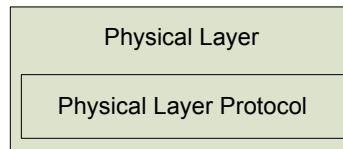


Figure 7—Physical Layer Protocol

- *Physical Layer Protocol*: Provides channel structure, frequency, power output, and modulation specifications for the forward and reverse links.

5.4.2.2.5 Security Functions

The protocols of the Security Function Sublayer are illustrated in Figure 8 and described in the text following.

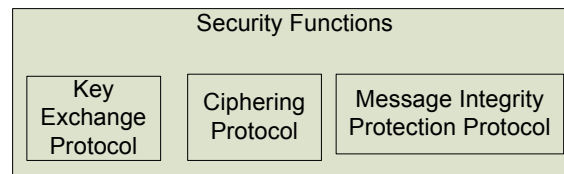


Figure 8—Security Protocols

- *Key Exchange Protocol*: Provides the procedures followed by the access network and the access terminal to generate security keys for message integrity protection and ciphering.
- *Message Integrity Protection Protocol*: Provides the procedures followed by the access network and the access terminal for integrity protection of signaling messages.
- *Ciphering Protocol*: Provides the procedures followed by the access network and the access terminal for ciphering traffic.

5.4.2.2.6 Connection Control Plane

The protocols of the Connection Control Plane Sublayer are illustrated in Figure 9 and described in the text following.

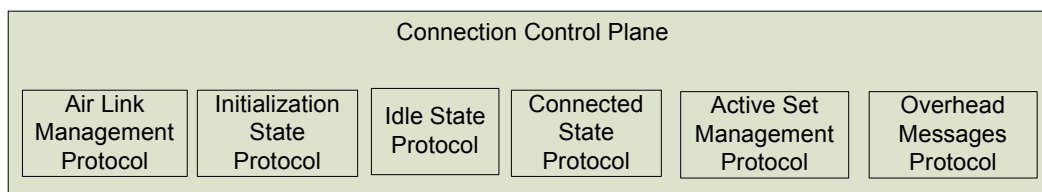


Figure 9—Connection Control Plane Protocols

The connection control plane contains the following six protocols used to manage connection state:

- *Air Link Management Protocol*: Provides the overall state machine management that an access terminal and an access network follow during a connection.
- *Initialization State Protocol*: Provides the procedures that an access terminal follows to acquire a network and that an access network follows to support network acquisition.
- *Idle State Protocol*: Provides the procedures that an access terminal and an access network follow when a connection is not open.
- *Connected State Protocol*: Provides the procedures that an access terminal and an access network follow when a connection is open.
- *Active Set Management Protocol*: Provides the means to maintain the air link between the access terminal and the access network.
- *Overhead Messages Protocol*: Provides broadcast messages containing information that is mostly used by Connection Control Plane protocols.

5.4.2.2.7 Session Control Plane

The protocol of the Session Control Plane Sublayer is illustrated in Figure 10 and described in the text following.

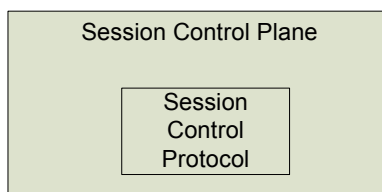


Figure 10—Session Control Plane Protocol

The session encapsulates all the state in the access network associated with an access terminal, whether connected, sleeping or even when powered down. The Session Control Plane contains one protocol as follows:

- *Session Control Protocol*: The Session Control Protocol provides negotiation and configuration of the protocols used in the session.

5.4.2.2.8 Route Control Plane

The protocol of the Route Control Plane Sublayer is illustrated in Figure 11 and described in the text following.

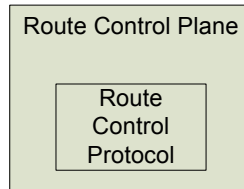


Figure 11—Route Control Plane Protocol

A route is an InUse instance of a protocol stack, used to communicate with a single access network. The route control plane contains one protocol as follows:

- *Route Control Protocol*: The Route Control Protocol performs creation, maintenance, and deletion of routes. The Route Control Protocol also performs management of access terminal identifier (ATI).

5.4.2.2.9 Broadcast-Multicast Service (BCMCS) Upper Layer Protocols

The protocols of the BCMCS Upper Layer Sublayer are illustrated in Figure 12 and described in the text following.

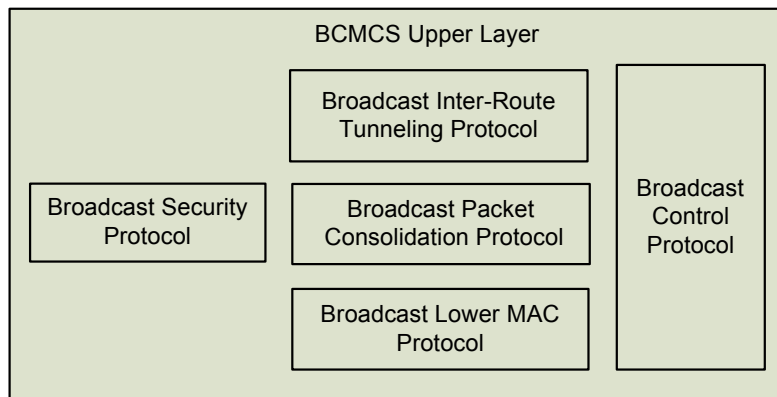


Figure 12—BCMCS Upper Layer Protocols

The following protocols in the BCMCS Upper Layer provide functions that offer broadcast-multicast service.

- *Broadcast MAC Protocol*: Defines the procedures followed by the access network to transmit, and by the access terminal to receive the BCMCS channel.
- *Broadcast Packet Consolidation Protocol*: Provides framing of BCMCS content packets and multiplexing of packets to be carried on the BCMCS channel.
- *Broadcast Inter-Route Tunneling Protocol*: Provides transport for packets from other routes.
- *Broadcast Security Protocol*: Provides ciphering of BCMCS content.
- *Broadcast Control Protocol*: Defines control procedures such as registration related to the BCMCS service.

5.4.2.2.10 Broadcast Physical Layer

The Broadcast Physical Layer is defined by the Physical Layer Protocol.

5.4.2.3 MIMO support

The air interface supports the following two modes of operation for the Lower MAC Sublayer and Physical Layer:

- Single-input-single-output (SISO) mode
- Multiple-input-multiple-output (MIMO) mode

The MIMO mode is divided into the following two sub-modes:

- Multiple codeword (MCW)
- Single codeword (SCW)

5.4.2.3.1 Access terminal requirements

All terminals shall support SISO mode. A MIMO-capable terminal shall support either the MIMO SCW or MCW sub-mode, and may operate, at a time, in either SISO mode or in one of the MIMO sub-modes. The MIMO mode requires at least two antennas at the access network and at least two antennas at the access terminal.

5.4.3 Channels

This standard refers to logical channels and physical channels. PHY (physical) channel names are in all uppercase, and MAC (logical) channels (typically on the reverse link) are displayed in all lowercase. A channel name consists of a prefix, an acronym, and a suffix. Table 1 shows the naming conventions for the channels in this standard. Special suffixes distinguish Pilot Channels (PICH) from Paging Channels.

Table 1—Channel naming conventions

Prefix	Acronym	Suffix
F- = Forward Link R- = Reverse Link (PHY) r- = Reverse Link (MAC)	(derived from capitalized letters of the channel name) ACK = acknowledgement ACQ = acquisition A = access Aux = auxiliary C = CDMA O = OFDMA	CH = Channel PICH = Pilot Channel (PHY)

There is no particular criterion used to define a channel. Some channels have unique modulations and encodings, whereas other channels are simply a few bits taken from a larger transmission unit such as the superframe preamble. In some cases, a channel corresponds to a control loop in the air interface. Channel names reference important sets of bits sent in this air interface, which are referenced repeatedly in other parts of the standard.

The forward link consists of the following parts:

- Guard carriers are used to mitigate interference with adjacent channels to meet regulatory requirements.
- Pilot channels are used for channel estimation.

- The Superframe Preamble carries slowly changing (tens of milliseconds) information.
- A PHY frame (within a Superframe, about 1 ms in length) is the basic unit of transport, carrying the following:
 - 1) Control information including grants and acknowledgements, etc.
 - 2) A Traffic Channel (containing bearer frames)

A complete listing of forward link channels is given in Table 216.

The reverse link consists of the following parts:

- Guard carriers are used to mitigate interference with adjacent channels to meet regulatory requirements
- Reverse CDMA Channels
 - 1) Pilot channels that are used for a power reference signal
 - 2) Access channel (used by ATs attempting to connect)
 - 3) Dedicated Control Channels (with logical subchannels)
 - 4) CDMA Data Channel
- Reverse OFDMA Channels
 - 1) Pilot channel that is used for channel estimation
 - 2) Dedicated Control Channels (with logical subchannels)
 - 3) Reverse link acknowledgement channels
 - 4) Reverse link data channels

A complete listing of reverse link channels is given in Table 212.

On the reverse link, the control channels are normally carried via CDMA, not OFDMA, and so the reverse link is a hybrid design. Two of the reverse-link channels have logical channels, as described as follows:

- a) Reverse CDMA Dedicated Control Channel (R-CDCCH)
 - r-cqich (channel quality indicator channel)
 - r-reqch (request channel)
 - r-pahch (power amplifier headroom channel)
 - r-psdch (spectral density indication channel)
- b) Reverse OFDMA Dedicated Control Channel (R-ODCCH)
 - r-mqich (mimo quality indicator channel)
 - r-cqich (channel quality indicator channel)
 - r-reqch (request channel)
 - r-sfch (subband feedback channel)
 - r-bfch (beam feedback channel)

5.4.4 Management Information Base

This standard includes the definition of a Management Information Base (MIB) module for managing the MAC and PHY. The objects in this MIB are defined using the mechanisms specified in the Structure of Management Information (SMI). The MIB module specified is compliant to SMIv2, which is described in IETF RFC 2578, IETF RFC 2579, and IETF RFC 2580. The MIB is defined in Clause 17.

5.4.5 Protocol Interfaces

In the following, the generic term *entity* is used to refer to the access terminal and the access network. This standard defines a set of interfaces for communications between protocols in the same entity and between a protocol executing in one entity and the same protocol executing in the other entity.

Protocols in this standard have the following five types of interfaces:

- Headers and/or Trailers are added by a protocol to payload received from an upper layer protocol, and are used for communications between a protocol executing in one entity and the same protocol executing in the other entity.
- Messages are generated by a protocol and are transported using the Signaling Protocol. Messages are used for communications between a protocol executing in one entity and the same protocol executing in the other entity. Messages can be transmitted using Reliable or Best Effort (unreliable) delivery.
- Commands are used by a protocol to obtain a service from another protocol within the same access network or access terminal.
- Indications are used by a protocol to convey information regarding the occurrence of an event to another protocol within the same access network or access terminal. Any protocol can register to receive these indications.
- Public data is used to share information in a controlled way between protocols. Public data is shared between protocols in the same sublayer, between protocols in different sublayers and layers, and between protocols in different routes.

Commands and indications are written in the form of *Protocol.Command* and *Protocol.Indication*. For example, *AccessChannelMAC.Activate* is a command activating the Access Channel MAC, and *IdleState.ConnectionOpened* is an indication provided by the Idle State Protocol that the connection is now open. When the context is clear, the *Protocol* part is dropped (e.g., within the Idle State Protocol, *Activate* refers to *IdleState.Activate*).

Unless specified otherwise, a protocol registers for indications from other protocols in the same route. Unless specified otherwise, a protocol sends/receives commands to/from other protocols in the same route.

Most protocols support the following two commands:

- *Activate*, which commands the protocol to transition from the Inactive state to some other state.
- *Deactivate*, which commands the protocol to transition to the Inactive state. Some protocols do not transition immediately to the Inactive state, due to requirements on orderly cleanup procedures.

Other common commands are *Open* and *Close*, which command protocols to perform session open/close or connection open/close related functions.

Commands are always written in the imperative form, since they direct an action. Indications are always written in the past tense since they notify of events that happened (e.g., *OpenConnection* for a command and *ConnectionOpened* for an indication).

Headers, trailers, and messages are binding on all implementations. Commands, indications, protocol data, and procedures that are internal to the access network or access terminal are used purely as a device for a clear and precise specification. Access terminals and access networks can be compliant with this standard while choosing a different implementation that exhibits identical behavior.

5.4.6 Protocol data

Each protocols owns and acts on parameters that are called protocol data. Protocol data can have various properties:

- *Public*: Protocol data is said to be public if the value of the data is visible outside the protocol. Value of public data may be read by other protocols within the same route as well as by protocols in other routes.
- *Static (Dynamic)*: Protocol data is said to be static if the data must take the same value across the same Protocol Type in different routes and personalities. Since the value of static data is the same across routes and personalities, all static data must be public. Protocol data is said to be dynamic if the data may take different values in different routes or personalities.
- *Local common data*: Local common data is static data that is maintained only by the access terminal.
- *Attribute*: Protocol data is called an attribute if the value of the data can be negotiated between the access terminal and the access network using configuration services provided by the InUse instance of the Session Control Protocol. Attributes may be either soft or hard committable. An attribute is said to be soft committable if changes to values of the data can take effect without closing the route. In contrast, an attribute is said to be hard committable if the route must be closed and re-opened for changes in data values to take effect.
- *Session data*: Protocol data is said to be session data if it is part of the air interface session that could be copied from one access network to another. All attributes are session data. Session data is conveyed between access network through Session State Information Records.

5.4.7 Protocol states

When protocols exhibit different behavior as a function of the environment (if a connection is opened or not, if a session is opened or not, etc.), this behavior is captured in a set of states and the events leading to a transition between states.

Unless otherwise specifically mentioned, the state of the access network refers to the state of a protocol engine in the access network as it applies to a particular access terminal. Since the access network communicates with multiple access terminals, multiple independent instantiations of a protocol will exist in the access network, each with its own independent state machine.

Unless otherwise specifically mentioned, the state of the access terminal refers to the state of a protocol engine in the access terminal as it applies to a particular access network. Since the access terminal communicates with multiple access networks, multiple independent instantiations of a protocol will exist in the access terminal (one in each route), each with its own independent state machine.

Unless otherwise specifically shown, the state transitions due to failure are not shown in the figures.

Typical events leading to a transition from one state to another are the receipt of a message, a command, an indication, or the expiration of a timer.

When a protocol is not functional at a particular time, the protocol is placed in a state that is usually called the Inactive state. This state is common for most protocols.

Other common states are Open, indicating that the session, connection, or route (as applicable to the protocol) is open and Close, indicating that the session, connection, or route is closed.

If a protocol has a single state other than the Inactive state, that state is always called the Active state. If a protocol has more than one state other than the Inactive state, all of these states are considered active, and are given individual names.

5.4.8 Protocol Set Identifier

The Protocol Set Identifier is used to identify the subtype of each of the protocols in the protocol stack. The Initial Protocol Set Identifier is used to identify the default protocol stack for each of the major revisions of the air interface standard. The access network includes values of the supported Initial Protocol Set Identifier in overhead messages. Multiple Protocol Set Identifiers may map to single Initial Protocol Set Identifier.

5.4.9 InUse and InConfiguration protocol transport instances

A protocol instance can be either an InUse instance or an InConfiguration instance. A protocol or transport instance is InUse if it is currently being used to communicate between the access terminal and the access network. Otherwise, a protocol or transport instance is InConfiguration.

Each protocol specifies procedures and messages corresponding to the InUse and InConfiguration protocol instances. In general, the InConfiguration protocol instances use the services of the Session Control Protocol to perform attribute configuration. Typically, non-configuration procedures and messages are processed by the InUse protocol instances.

An InConfiguration instance of each protocol is created by the Session Control Protocol once the session configuration is initiated (e.g., in the Basic Session Control Protocol this occurs upon entering the AT Initiated state). The initialization procedures for an InConfiguration protocol instance are invoked upon creation of the InConfiguration protocol instance. InConfiguration protocol instances can be changed by the Session Control Protocol.

The configured InConfiguration instances can be stored. Stored protocol stacks configurations are called Personalities. The access terminal and access network can store multiple Personalities as part of the session (see 5.4.10 for a discussion of sessions).

The access terminal and the access network determine which Personality to use for every route. Once the access terminal and access network agree upon using a Personality, then that Personality is Committed to the InUse instance. Each InUse protocol defines a set of Hard Commit and Soft Commit procedures. The Commit procedures for a protocol are invoked by the InUse instance of the Session Control Protocol. Commit procedures are used to commit a stored Personality to the InUse instance. The initialization procedures for an InUse protocol instance are invoked upon creation of the InUse protocol instance.

5.4.10 Sessions and connections

A session refers to a shared state between the access terminal and the access network. This shared state stores one or more Personalities. Each Personality contains the protocols and protocol configurations that were negotiated and are used for communications between the access terminal and the access network.

Other than to open a session, an access terminal cannot communicate with an access network without having an open session.

A connection is a particular state of the air-link in which the access terminal is assigned dedicated communication channels.

During a single session the access terminal and the access network can open and close a connection multiple times.

5.4.11 Security

The air interface supports security functions, which can be used for integrity protection of signaling messages, encryption of signaling messages, and other traffic transported between the access terminal and the access network. Unless specified otherwise, all signaling messages shall be integrity protected.

5.4.12 Support for Quality of Service (QoS)

QoS parameters are established and maintained at upper layers on the protocol stacks (i.e., layers above the MAC layer) on a per IP flow (as defined in IETF RFC 2216) basis, between the access terminal and various infrastructure entities that participate in the establishment and delivery of services. The exact functionality and the employed protocols are outside the scope of this standard. From an air interface point of view, QoS related information is carried transparently by air interface messages between the AT and the AN, encapsulated in profiles. A profile is a data structure associated with an IP flow, that contains one or more QoS attribute sets. A QoS attribute set specifies either a standard setting of values for QoS parameters known as a flow profile and identified in 3GPP2 C.R1001-F v1.0., or an ad-hoc setting of values for specific QoS parameters, including traffic class, peak rate, bucket size, token rate, maximum latency, maximum IP packet loss rate, packet size, and sensitivity to jitter.

Each QoS attribute set is uniquely identified by an assigned ID, known as QoS_ATTRIBUTE_SET_ID. As specified in the Basic QoS Management protocol (see 7.2), the AN and AT synchronize with each other on the use of the QoS attribute sets by setting the selected QoS attribute set ids within the ProfileValue field transmitted in the air interface messages.

5.4.13 Route-Tunneling interface

Systems deploying Wideband mode may support a Route-Tunneling interface, to allow for rapid and efficient transfer of control and data between access networks. The Route-Tunneling interface's control plane is assumed to be able to set up connections between at least two access networks, including support for the tunnel envisioned in the Inter-Route Tunneling Protocol (IRTP) of the Application Sublayer (see 6.3). The Route-Tunneling interface's user plane is assumed capable of efficient transport of the IRTP payload, via the connection established by the control plane.

This standard does not require or specify any particular Route-Tunneling interface.

5.4.14 Message information

Each message definition contains information regarding channels on which the message can be transmitted, whether the message requires RLP reliable or best-effort delivery, the addressing modes applicable to the message, and whether AuthTag is required in the message. This information is provided in the form of a table, an example of which is given in Figure 13.

Channels			FTC	RTC	RLP		Best Effort
Addressing			Unicast		AuthTag	Required	

Figure 13—Sample message information

The following values are defined:

- Channels: This information field indicates the Physical Layer Channels on which this message can be transmitted. The sender of the message shall send the message only on the Physical Layer channel(s) indicated by this information field. Values are

- FTC for Forward Unicast Traffic Channel
- F-BCMCSCH for Forward Broadcast Multicast Services Channel
- RTC for Reverse Traffic Channel
- b) RLP: Radio Link Protocol requirements. The sender of the message shall send the message only using the RLP in the mode(s) indicated by this information field. Values are
 - Best Effort: the message is sent once and is subject to erasure
 - Reliable: erasures are detected and the message is retransmitted one or more times, if necessary
- c) Addressing: Addressing modes for the message. The sender of the message shall send the message only with an address type(s) indicated by this information field. Values are
 - Broadcast: if a broadcast address can be used with this message
 - Unicast: if a unicast address can be used with this message
- d) AuthTag: AuthTag requirements. The sender of the message shall send the message with AuthTag requirements indicated by this information field. Values are
 - Required: if this message shall not be sent without an AuthTag
 - Not Required: if an AuthTag is not required with this message
 - Required when key is available: if an AuthTag shall be sent with this message when Message Integrity Key for the channel carrying this message is available (e.g., FACMIKey[KeyIndex] is not set to NULL). See 10.3.5.1.6.

5.4.15 Notation

$A[i]$	The i^{th} element of array A. The first element of the array is $A[0]$.
$\langle e_1, e_2, \dots, e_n \rangle$	A <i>structure</i> with elements ' e_1 ', ' e_2 ', ..., ' e_n '. Two structures $E = \langle e_1, e_2, \dots, e_n \rangle$ and $F = \langle f_1, f_2, \dots, f_m \rangle$ are equal if and only if ' m ' is equal to ' n ' and e_i is equal to f_i for $i = 1, \dots, n$. Given $E = \langle e_1, e_2, \dots, e_n \rangle$ and $F = \langle f_1, f_2, \dots, f_m \rangle$, the assignment " $E = F$ " denotes the following set of assignments: $e_i = f_i$, for $i = 1, \dots, n$.
$S.e$	The member of the structure ' S ' that is identified by ' e '.
$M[i:j]$	Bits i^{th} through j^{th} inclusive ($i \geq j$) of the binary representation of variable M. $M[0:0]$ denotes the least significant bit of M.
	Concatenation operator. $(A B)$ denotes variable A concatenated with variable B.
$\times$	Indicates multiplication.
$\lfloor x \rfloor$	Indicates the largest integer less than or equal to x: $\lfloor 1.1 \rfloor = 1, \lfloor 1.0 \rfloor = 1$.
$\lceil x \rceil$	Indicates the smallest integer greater or equal to x: $\lceil 1.1 \rceil = 2, \lceil 2.0 \rceil = 2$.
$ x $	Indicates the absolute value of x: $ -17 = 17, 17 = 17$.
$\oplus$	Indicates exclusive OR (modulo-2 addition).
$\otimes$	Indicates bitwise logical AND operator.
$\min(x, y)$	Indicates the minimum of x and y.
$\max(x, y)$	Indicates the maximum of x and y.

$x \bmod y$	Indicates the remainder after dividing x by y : $x \bmod y = x - (y \times \lfloor x/y \rfloor)$.
x^y	Indicates the result of x raised to the power y , also denoted as x^y .
x^y	Indicates the result of x raised to the power y , also denoted as x^y .
$N!$	Indicates the factorial of N , i.e., $N! = 1 \times 2 \times 3 \times \dots \times N$, for $N \geq 0$.
Rounding	Indicates the nearest integer: 1 for 1.4, -1 for -1.4 , 2 for 1.5, and -2 for -1.5 .
${}^nC_r$	$n! / [(n-r)! \times r!]$, for $n \geq r \geq 0$.
${}^nP_r$	$n! / (n-r)!$, for $n \geq r \geq 0$.
NN	Indicates a hexadecimal pair of digits in an attribute name that associates it with a particular RLP stream (allowable values of 00 to 1F inclusive).
KK	Indicates a hexadecimal pair of digits in an attribute name that associates it with a particular QoS reservation (allowable values of 00 to FF inclusive).

Unless otherwise specified, the format of field values is unsigned binary.

Unless indicated otherwise, this standard presents numbers in decimal form. Binary numbers are distinguished in the text by the use of single quotation marks. Hexadecimal numbers are distinguished by the prefix '0x'.

Unless specified otherwise, each field of a packet shall be transmitted in sequence such that the most significant bit (MSB) is transmitted first and the least significant bit (LSB) is transmitted last. The MSB is the left-most bit in the figures in this standard. If there are multiple rows in a table, the top-most row is transmitted first. If a table is used to show the sub-fields of a particular field or variable, the top-most row consists of the MSBs of the field. Within a row in a table, the left-most bit is transmitted first. Notations of the form "repetition factor of N " or "repeated N times" mean that a total of N versions of the item are used.

Characteristics of parameter attributes are specified according to a graphic convention. The attribute is specified and the characteristic value of the attribute is indicated in an adjacent cell of a two cell table. An example of such a graphic is shown here as follows:

Commit	Soft	Scope	Dynamic

5.4.16 Malfunction detection

The access terminal shall have a malfunction timer that is separate from and independent of all other functions and that runs continuously whenever power is applied to the transmitter of the access terminal. Sufficient reset commands shall be interspersed throughout the access terminal logic program to ensure that the timer never expires as long as the proper sequence of operations is taking place. If the timer expires, a malfunction shall be assumed and the access terminal should be inhibited from transmitting. The maximum time allowed for expiration of the timer is 2 s.

5.4.17 Revision number

Access terminals and access networks complying with the requirements of this standard shall set their revision number to 0x01.

5.5 System overview of 625k-MC (625kiloHertz-spaced MultiCarrier) Mode

5.5.1 Scope

This standard defines the physical layer, medium access control, and the signaling layer for 625k-MC, which is an enhancement of ATIS-0700004.2005, the scope of which is presented in 1.1 of that document. For brevity, references to the baseline specification: ATIS 0700004.2005, High Capacity-Spatial Division Multiple Access (HC-SDMA) Radio Interface Standard, September 2005 will be denoted as “ATIS 0700004.2005” in this standard. Unless otherwise specified in this document, the requirements of ATIS-0700004.2005 shall apply to the 625k-MC mode of IEEE Std 802.20.

5.5.2 Architecture reference model

The architecture reference model for 625k-MC Mode is presented in Figure 14. The reference model includes the air interface between the user terminal and the base station. The basic specifications of the 625k-MC shall be those specified in ATIS-0700004.2005 and are referred to as *basic protocols*. The 625k-MC mode includes additional specifications that supplement the basic protocols, which shall be as specified herein and shall be referred to as *enhanced protocols* with reference to Base Specifications of ATIS-0700004.2005.

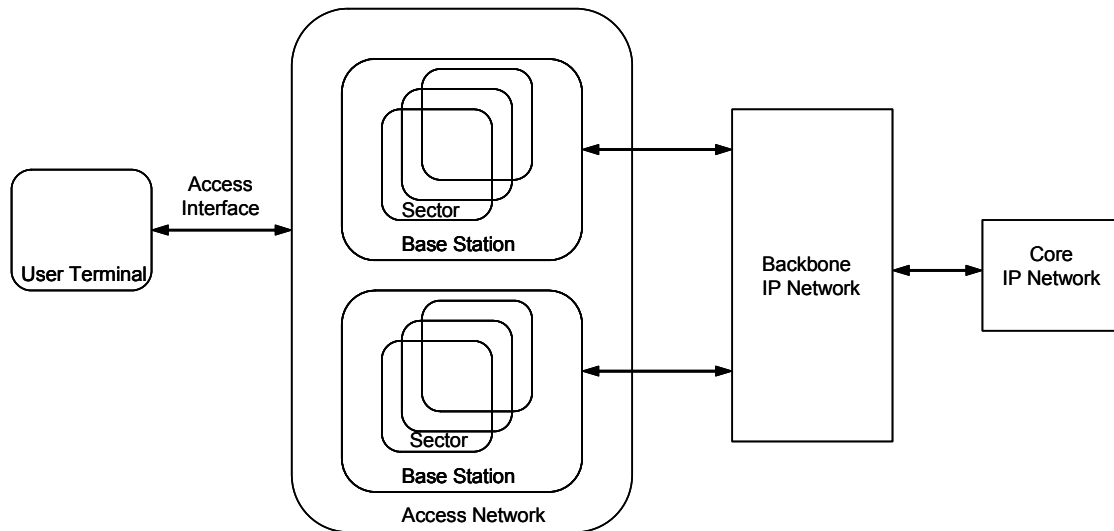


Figure 14—625k-MC Architecture reference model

The functional units of the reference architecture in Figure 14 are defined in Clause 3 and are included below for convenience of the reader:

- *access network*: The network equipment (a collection of base stations) providing IP layer connectivity between an IP network (typically the Internet) and the user terminals.
- *base station (BS)*: The device in the access network that communicates over the air interface, via one or more sectors, with the user terminals. Base stations coordinate the management of the air interface attributes.
- *user terminal (UT)*: A device providing data connectivity to an end user device (EUD) user. A user terminal may be connected to a computing device, such as a laptop personal computer, or it may be a self-contained data device, such as a personal digital assistant.

- *sector*: One set of physical layer channels transmitted between base station and the user terminals within a given frequency assignment.
- *downlink (DL)*: The radio connection with the BS transmitting and the UT receiving. The term is used interchangeably with the term *forward link* or *FL*.
- *uplink (UL)*: The radio connection with the UT transmitting and the BS receiving. The term is used interchangeably with the term *reverse link* or *RL*.

5.5.3 Conventions

The terminologies and conventions used in this standard shall be as specified in 1.3 of ATIS-0700004.2005.

For the 625k-MC mode, unless specified otherwise, each field of a packet shall be transmitted in sequence such that the least significant bit (LSB) is transmitted first and the most significant bit (MSB) is transmitted last. The LSB is the left-most bit in the figures ATIS-0700004.2005 and in the clauses in this document specifying the 625-kHz mode.

In procedural descriptions throughout the text describing operations in the 625k-MC mode, the user terminal (UT) and base station (BS) shall perform the indicated steps in the sequence given, unless otherwise stated.

5.5.4 625k-MC application overview

The 625k-MC application overview is as specified in 1.4 of ATIS-0700004.2005.

5.5.5 625k-MC Protocol overview

625k-MC mode's RF/PHY/MAC/RLC/DLC specifications are based on the L0/L1/L2/L3 specifications defined in ATIS-0700004.2005. The 625k-MC PHY/MAC/RLC/DLC Protocol Reference model is as shown in Figure 15.

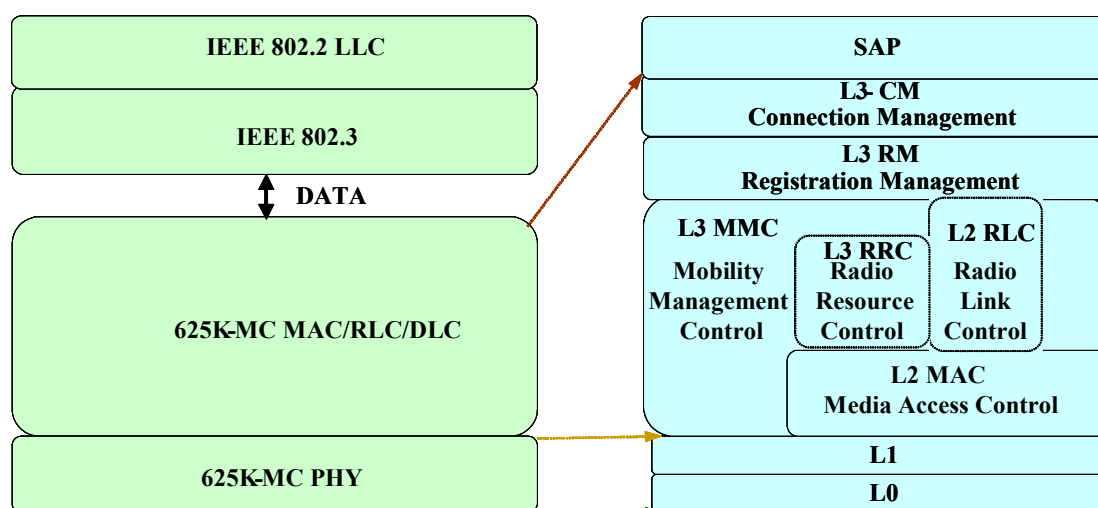


Figure 15—625k-MC PHY/MAC/RLC/DLC Protocol reference

The 625k-MC system enables end-to-end IP-over-PPP (Point-to-Point Protocol) connectivity between the service providers and their customers who are the end users.

Data traffic from an end-user device attached to a user terminal (UT) or embedded device within a user terminal is transmitted encapsulated in IEEE 802.2 Logical Link Control (LLC)/IEEE 802.3 above the 625k-MC MAC (L2 MAC Media Access Control in ATIS-0700004.2005).

Data traffic forwarded by the base station (BS) or an application within the BS to the UT or an end-user device attached to the UT is encapsulated in IEEE 802.2 LLC/802.3 above the 625k-MC MAC (L2 MAC Media Access Control in ATIS-0700004.2005).

625k-MC L3 Management and Control messages (Clause 20 and Clause 21) exchanged between UT and BS are not encapsulated in IEEE 802.2 LLC/IEEE 802.3 across the Air Interface.

To support the interaction and communication between protocol layers, for example between L3 and IEEE 802.2 LLC/Ethernet layers, various primitives are defined in Clause 22.

The 625k-MC Physical layer (PHY) specifications are based on the definitions of L1 (Layer 1) and L0 (Layer 0) protocols of ATIS-0700004.2005. The 625k-MC Medium Access Control (MAC) specifications are based on the definitions of the L2-MAC (Layer 2-MAC) protocol of ATIS-0700004.2005. The 625k-MC Data Link Control (DLC) and radio link Control (RLC) specifications are based on the definitions of L2-RLC (Layer 2-Radio Link Control), Layer L3 (L3-RRC, L3-MMC, L3-CM, L3-RM) of ATIS-0700004.2005.

5.5.5.1 625k-MC Protocol features

The basic protocol features of 625k-MC shall be the protocol features that are defined in 1.5.1 of ATIS-0700004.2005.

5.5.5.2 625k-MC Protocol reference model and interfaces

The 625k-MC protocol reference model and communication interface between protocol layers shall be as defined in 1.6 of ATIS-0700004.2005.

6. Services Sublayer

6.1 Introduction

The Services Sublayer consists of following protocols:

- Basic Signaling Protocol
- Basic Inter-Route Tunneling Protocol
- Basic Robust Header Compression (ROHC) Support Protocol
- Basic EAP Support Protocol

As shown in Figure 7, other protocols may also produce Services Sublayer Packets to be carried over the Radio Link Sublayer. Examples of these include IEEE 802.1Q C-VLAN frames (see IEEE Std 802.1Q) and IEEE 802.3 Ethernet PDUs (see IEEE Std 802.3).

6.2 Basic Signaling Protocol

6.2.1 General overview

Protocols in each layer use the Basic Signaling Protocol (BSP) to exchange messages. BSP is also used by application specific control messages.

BSP provides a one or two octet header that defines the Type of the protocol and the protocol instance (i.e., InConfiguration or InUse) with which the message is associated. The BSP uses the header to route the message to the appropriate protocol instance.

BSP is a message-routing protocol and routes messages to protocols specified by the Type field provided in the BSP header.

Protocol type is assigned to the InUse protocol instance as well as the InConfiguration instance of each protocol.

The actual protocol indicated by the Type is negotiated during session set-up. For example, Type 0x07 is associated with the Reverse Control Channel MAC Protocol. The specific Reverse Control Channel MAC Protocol used (and, therefore, the Reverse Control Channel MAC protocol generating and processing the messages delivered by BSP) is negotiated when the session is set up.

The remainder of the message following the Type field (BSP header) is processed by the protocol specified by the Type.

6.2.2 Primitives and public data

6.2.2.1 Commands

This protocol does not define any commands.

6.2.2.2 Return indications

This protocol returns the following indications:

- *ReliableMessageDeliveryFailed*
- *ReliableSignalingStreamReset*

6.2.2.3 Local common data

This protocol does not define any local common data.

6.2.2.4 Public data

This protocol shall make the following data public:

- All data defined as Static Attribute Data, Static Non-Attribute Data, and local common data.

6.2.3 Protocol data unit

The protocol data unit for this protocol is a BSP packet. Each BSP packet consists of one message sent by a protocol using BSP.

The protocol constructs a BSP packet by adding the BSP header (see 6.2.6.7) in front of the payload. The structure of the BSP packet is shown in Figure 16.

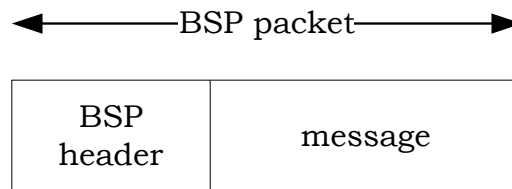


Figure 16—BSP packet structure

6.2.4 General signaling requirements

6.2.4.1 General requirements

The following requirements are common to all protocols that carry messages using BSP and that provide for message extensibility. The access terminal and the access network shall abide by the following rules when generating and processing any signaling message carried by BSP:

- Messages shall always be an integer number of octets in length; and, if necessary, include a Reserved or Padding field in the message to make them so. The receiver shall ignore the value of the Reserved or Padding fields.
- The first field of the message shall be transmitted first. Within each field, the most significant bit of the field shall be transmitted first.
- Message identifiers for messages that cannot be sent using broadcast addressing shall be unambiguous for each protocol (Type, Subtype) pair.
- Message identifiers for messages that may be sent using broadcast addressing shall be unambiguous for each protocol Type.
- For future revisions, the transmitter shall add new fields only at the end of a message (excluding any trailing Reserved or Padding field). The transmitter shall not add fields if their addition makes the parsing of previous fields ambiguous for receivers.
- The receiver shall discard all unrecognized messages.
- The receiver shall discard all unrecognized fields.

- The receiver shall discard a message if any of the fields in the message is set to a value outside the defined field range, unless the receiver is specifically directed to ignore this field. A field value is outside the allowed range if a range was specified with the field and the value is not in this range, or the field is set to a value that is defined as invalid. Values that are designated as reserved are considered to be outside the allowed range.

6.2.5 Procedures and messages for the InConfiguration instance of the protocol

6.2.5.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

6.2.5.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

6.2.5.3 Message formats

This protocol does not define any messages.

6.2.6 Procedures and messages for the InUse instance of the protocol

6.2.6.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in 15.1.2.

6.2.6.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

6.2.6.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

6.2.6.4 Procedures

The BSP receives messages for transmission from multiple protocols. The BSP shall add the BSP header to each message and forward it for transmission to RLP.

The BSP receives messages from RLP. The BSP shall route these messages to their associated protocols according to the value of the Type field in the BSP header. The BSP shall route the message to the instance of the protocol identified by the Type field.

If the BSP receives *RadioLinkNN.ReliablePacketDeliveryFailed* indication, where *NN* corresponds to an RLP instance associated with the BSP, then the BSP shall return a *ReliableMessageDeliveryFailed* indication.

If BSP receives *RadioLinkNN.ReliableStreamReset* indication, where *NN* corresponds to RLP instance associated with the BSP, then the BSP shall return a *ReliableSignalingStreamReset* indication.

6.2.6.5 Type definitions

Protocol Type values assigned to protocols are specified in 16.1.

6.2.6.6 Messages

No messages are defined for the InUse instance of the protocol.

6.2.6.7 BSP header

The BSP shall place the following header (see Table 2) in front of every message that it sends:

Table 2—BSP header

Field	Length (bits)
ProtocolType	8 or 16

ProtocolType This field shall be set to the ProtocolType value for the protocol (as defined in 16.1) associated with the encapsulated message. Protocol Types are allocated in such a way that if the first bit is a '0', then this is an 8-bit Protocol Type, and if the first bit is a '1', then this is a 16-bit Protocol Type.

6.2.7 Interface to other protocols

6.2.7.1 Commands

This protocol does not issue any commands.

6.2.7.2 Indications

This protocol registers to receive the following indications:

- *RadioLinkNN.ReliablePacketDeliveryFailed*
- *RadioLinkNN.ReliableStreamReset*

6.2.8 Configuration attributes

This protocol does not define any simple attributes or complex attributes.

6.2.9 Non-attribute data

This protocol does not define any non-attribute data.

6.2.10 Session State Information

The Session State Information record (see Clause 15) consists of parameter records. All configuration attributes and non-attribute data are Session State Information records. This protocol does not define additional parameter records in addition to the configuration attributes for this protocol.

6.3 Basic Inter-Route Tunneling Protocol

6.3.1 Overview

The Inter-Route Tunneling Protocol performs the following function:

- Tunneling of data belonging to different routes. The Inter-Route Tunneling Protocol Header indicates the route to which the payload belongs. The Inter-Route Tunneling Protocol allows one route to carry payload bound for another route, including payload bound for its route.

A route consists of an InUse protocol stack associated with an access network.

At the transmitter, the Inter-Route Tunneling Protocol receives packets for transmission from the Route Protocol of another route or from the Route Protocol of the same route (see Figure 17). The Inter-Route Tunneling Protocol adds an Inter-Route Tunneling Protocol Header to the received packet to identify the destination route and delivers this packet to the Radio Link Protocol. The Route Tunneling interface in Figure 17 refers to a generic communication link between ANs. Further definition of the Route Tunneling interface is beyond the scope of this standard.

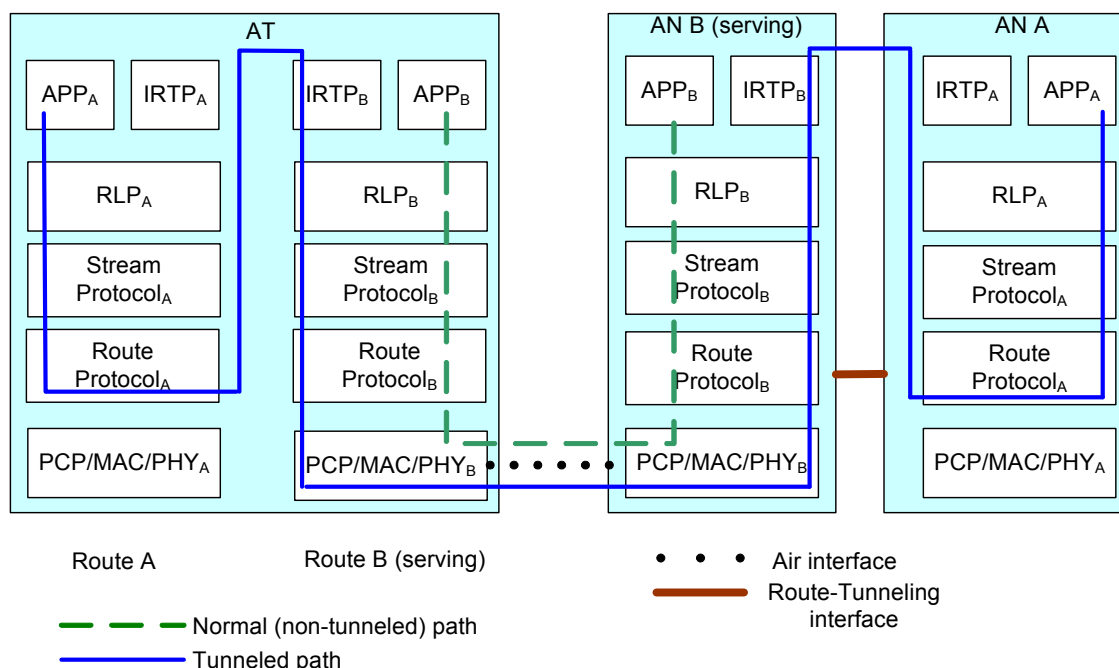


Figure 17—Data path for normal and tunneled packets

At the receiver, the Inter-Route Tunneling Protocol receives packets from the Radio Link Protocol. The Inter-Route Tunneling Protocol removes the Inter-Route Tunneling Protocol Header and delivers the packet to the Route Protocol of the corresponding route.

The Inter-Route Tunneling Protocol may receive packets from the Route Protocol of its route for further fragmentation at RLP.

Figure 18 illustrates various configurations of the Inter-Route Tunneling Protocol Headers. The usage of PilotID and ANID is specified in 9.2.2.2.1 and 16.2.1, respectively.

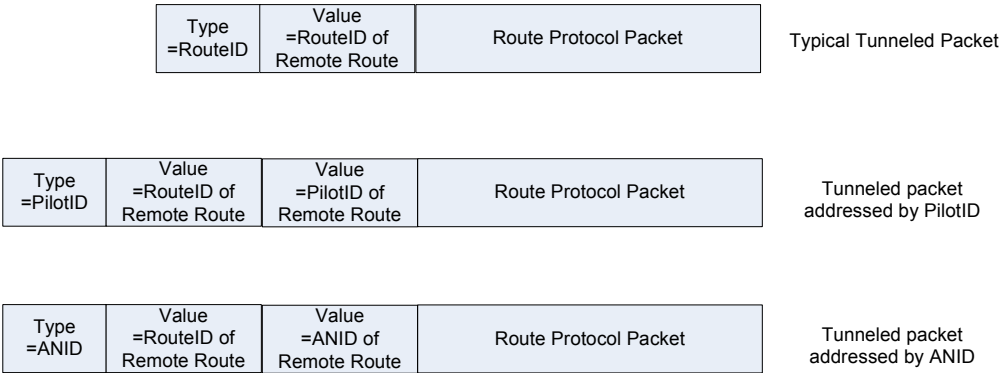


Figure 18—Example Inter-Route Tunneling Protocol Headers

6.3.2 Primitives, local common data, and public data

6.3.2.1 Commands

This protocol does not define any commands.

6.3.2.2 Return indications

This protocol does not return any indications.

6.3.2.3 Local common data

This protocol does not define any local common data.

6.3.2.4 Public data

This protocol shall make the following data public:

- All data defined as Static Attribute, Static Non-Attribute Data, and local common data.

6.3.3 Protocol data unit

The protocol data unit for this protocol is an Inter-Route Tunneling Protocol Packet. An Inter-Route Tunneling Protocol Packet consists of an Inter-Route Tunneling Protocol Payload and an Inter-Route Tunneling Protocol Header (see Figure 19).

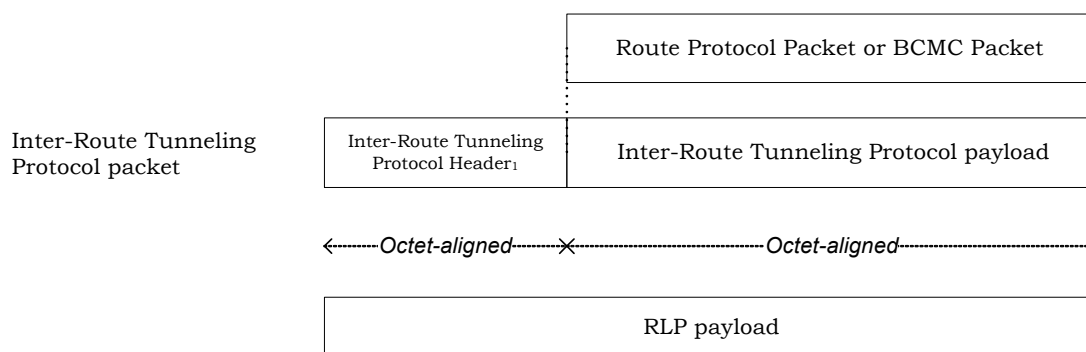


Figure 19—Inter-Route Tunneling Protocol Packet

6.3.4 Procedures and messages for the InConfiguration instance of the protocol

6.3.4.1 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

When configuring the *ReservationKKIRTPStreamRev* attribute for a Reservation *KK*, the access terminal shall not set it to a Stream that is not defined as a Inter-Route Tunneling Protocol stream by the Radio Link Protocol.

6.3.4.1.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

6.3.4.2 Message formats

This protocol does not define any messages.

6.3.5 Procedures and messages for the InUse instance of the protocol

6.3.5.1 Procedures

6.3.5.1.1 Protocol initialization

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in 15.1.2.

6.3.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

6.3.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

6.3.5.1.4 Inter-Route Tunneling Protocol Header

The Inter-Route Tunneling Protocol Header has the format shown in Table 3.

Table 3—Inter-Route Tunneling Protocol Header

Field	Length (bits)
HeaderType	1 or 4
If HeaderType= '0'	
RouteID	7
If HeaderType= '1000'	
RouteIDIncluded	1
RouteID	7 ^a
PilotID	10
If HeaderType= '1001'	
RouteIDIncluded	1
RouteID	7 ^a
Reserved	3 or 4
ANID	64
If HeaderType='1111'	
SpecialRouteID	4
Padding	0–7 (as needed)

^aThis field may be absent (see field description for conditions).

HeaderType

The sender shall set this field as specified in Table 4 to indicate the type of Inter-Route Tunneling Protocol Header.

Table 4—Header Type values

HeaderType (binary)	Type of Inter-Route Tunneling Protocol Header
0	RouteID Header
1000	PilotID Header
1001	ANID Header
1111	SpecialRouteID Header
Other values	Reserved

RouteID

The sender shall set this field to the RouteID assigned to the route to which this packet is destined.

If the HeaderType field is set to '1000', the sender shall include the following three-field record:

RouteIDIncluded	The access terminal shall set this field to '1'. The access network shall set this field to '0' if the packet being tunnelled is a broadcast overhead message; otherwise, the access network shall set this field to '1'.
RouteID	The sender shall omit this field if the RouteIDIncluded field is set to '0'; otherwise, the sender shall include this field and set it to the RouteID assigned to the route to which this packet is destined.
PilotID	The sender shall set this field to the pilot identifier of a pilot belonging to the access network to which this packet is destined.

If the HeaderType field is set to '1001', the sender shall include the following four-field record:

RouteIDIncluded	The access terminal shall set this field to '1'. The access network shall set this field to '0' if the packet being tunnelled is a broadcast overhead message; otherwise, the access network shall set this field to '1'.
RouteID	The sender shall omit this field if the RouteIDIncluded field is set to '0'; otherwise, the sender shall include this field and set it to the RouteID assigned to the route to which this packet is destined.
Reserved	If the RouteIDIncluded field is set to '0', then length of this field shall be 3 bits; otherwise, length of this field shall be 4 bits. The receiver shall ignore these bits.
ANID	The sender shall set this field to the access network identifier (see 16.2.1) of the access network to which this packet is destined.

If the HeaderType field is set to '1111', the sender shall include the following one-field record:

SpecialRouteID	The sender shall set this field to the special Route Identifier, as specified in Table 5, corresponding to the route to which this packet is destined.

Table 5—SpecialRouteID Values

RouteID (binary)	RouteID usage
0000	BCMC Route
0001–1111	Reserved

Padding	The sender shall include zero to seven bits to make this Inter-Route Tunneling Protocol Header octet-aligned. The sender shall set these bits to 0. The receiver shall ignore these bits.

6.3.5.1.5 Command processing

6.3.5.1.5.1 Activate

None.

6.3.5.1.5.2 Deactivate

None.

6.3.5.1.6 Inter-Route Tunneling Protocol Stream Mapping procedures

If the State public data of the Route Control Protocol is *WaitingToOpen*, the access terminal shall map all packets to Stream 0x04; otherwise, the access terminal shall perform the following:

- The access terminal shall map all packets not associated with a *ReservationKK* (i.e., packets generated by applications other than IP or ROHC) to Stream 0x04.
- The access terminal should map all packets associated with a *ReservationKK* to the Stream as specified by *ReservationKKIRTPStreamRev*.

If the State public data of the Route Control Protocol is *WaitingToOpen*, the access network shall map all packets to Stream 0x04.

6.3.5.1.7 Transmission procedures

6.3.5.1.7.1 Access terminal procedure

If the Inter-Route Tunneling Protocol receives a Route Protocol Packet from the Route Protocol of a route, it shall perform the following:

- a) The access terminal shall satisfy the following requirements in setting the fields of the Inter-Route Tunneling Protocol header:
 - If the State public data of the Route Control Protocol is *WaitingToOpen*, the access terminal shall not use the following *HeaderTypes*:
 - 1) *HeaderType* of '1000'
 - 2) *HeaderType* of '1001' unless it is in response to a *RouteCreate* message
 - Otherwise, the access terminal shall not use *HeaderType* of '0' for a destination route if the access terminal has not received the acknowledgement for the *RouteMap* information sent to this access network containing the *ANID* to *RouteID* mapping for this destination route.
- b) The Inter-Route Tunneling Protocol shall add an Inter-Route Tunneling Protocol Header to identify the destination route.
- c) The Inter-Route Tunneling Protocol shall deliver the Inter-Route Tunneling Protocol Packet to the Radio Link Protocol of its route.

6.3.5.1.7.2 Access network procedures

If the Inter-Route Tunneling Protocol receives a Route Protocol Packet from the Route Protocol of a route, it shall perform the following:

- a) The access network shall satisfy the following requirements in setting the fields of the Inter-Route Tunneling Protocol header:
 - The access network shall not use *HeaderType* '1001' (*ANID*) in the Inter-Route Tunneling Protocol Header until the *RouteMap* information is received.
- b) The Inter-Route Tunneling Protocol shall add an Inter-Route Tunneling Protocol Header to identify the destination Route.
- c) The Inter-Route Tunneling Protocol shall deliver the Inter-Route Tunneling Protocol Packet to the Radio Link Protocol of its route.

6.3.5.1.8 Reception procedures

6.3.5.1.8.1 Access terminal procedures

If the Inter-Route Tunneling Protocol receives an Inter-Route Tunneling Protocol Packet from the Radio Link Protocol of its route, it shall perform the following:

- a) The Inter-Route Tunneling Protocol shall remove the Inter-Route Tunneling Protocol Header to produce an Inter-Route Tunneling Protocol Payload.
- b) The Inter-Route Tunneling Protocol shall perform the following:
 - If the HeaderType field of the header is set to '0', the Inter-Route Tunneling Protocol shall deliver the Inter-Route Tunneling Protocol Payload to the Route Protocol of the route identified in the RouteID field of the header.
 - If the HeaderType field of the header is set to '1000', the Inter-Route Tunneling Protocol shall deliver the Inter-Route Tunneling Protocol Payload to the Route Protocol associated with the pilot identified in the PilotID field of the header, along with the PilotID value.¹⁰
 - If the HeaderType field of the header is set to '1001', the Inter-Route Tunneling Protocol shall deliver the Inter-Route Tunneling Protocol Payload to the Route Protocol associated with the route associated with the received ANIDmsb.

6.3.5.1.8.2 Access network procedures

If the Inter-Route Tunneling Protocol receives an Inter-Route Tunneling Protocol Packet from the Radio Link Protocol of its route, it shall perform the following:

- a) The Inter-Route Tunneling Protocol shall remove all Inter-Route Tunneling Protocol Headers present to produce an Inter-Route Tunneling Protocol Payload.
- b) If the HeaderType field of the header is set to '0', the access network shall perform the following:
 - If the access network does not recognize the RouteID field of the header:
 - 1) If the State public data of the Route Control Protocol is WaitingToOpen, the Inter-Route Tunneling Protocol shall deliver a copy of the Inter-Route Tunneling Protocol Payload to the Route Protocol of the SessionAnchor Route.
 - 2) Otherwise, the Inter-Route Tunneling Protocol shall issue a QueryRouteMap command.
 - Otherwise, the Inter-Route Tunneling Protocol shall deliver a copy of the Inter-Route Tunneling Protocol Payload to the Route Protocol of the route identified in the RouteID field of the header.
- c) If the HeaderType field of the header is set to '1000', the Inter-Route Tunneling Protocol shall deliver the Inter-Route Tunneling Protocol Payload to the Route Protocol of the access network to which the sector identified in the PilotID field of the header belongs.
- d) If the HeaderType field of the header is set to '1001', the Inter-Route Tunneling Protocol shall deliver the Inter-Route Tunneling Protocol Payload to the Route Protocol of the access network corresponding to the received access network identifier.

6.3.5.2 Type definitions

Protocol Type values assigned to protocols are specified in Clause 16.

6.3.5.3 Message formats

This protocol does not define any messages.

¹⁰Since separate RLP queue is maintained for each sector of a route, the PilotID information is necessary to deliver this packet to the correct RLP queue.

6.3.6 Interface to other protocols

6.3.6.1 Commands sent

This protocol issues the following command:

— *RouteControl.QueryRouteMap*

6.3.6.2 Indications

This protocol does not register to receive any indications.

6.3.7 Configuration attributes

6.3.7.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 6. The access terminal and the access network shall use as defaults the values in Table 6 typed in ***bold italics***.

Table 6—Configurable values

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x01 <i>KK</i> <i>KK</i> is the two-digit hexadecimal Reservation Label.	Reservation <i>KK</i> IRTPStreamRev <i>KK</i> is the two-digit hexadecimal Reverse Link ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters.	Soft/ Dynamic	<i>0x06</i>	Stream 0x06 is associated with this Reverse ReservationLabel.
			0x04–0x05, 0x07–0x1e	Stream associated with this Reverse ReservationLabel.
			All other values	Reserved

6.3.7.2 Complex attributes

This protocol does not define any complex attributes.

6.3.8 Non-attribute data

This protocol does not define any non-attribute data.

6.3.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records. All configuration attributes and non-attribute data are Session State Information records. This protocol does not define additional parameter records.

6.4 Basic Robust Header Compression Support Protocol

6.4.1 Overview

The ROHC Support Protocol performs the following functions:

- Negotiate ROHC attributes
- Sending and Receiving ROHC protocol packets

6.4.2 Primitives, local common data, and public data

6.4.2.1 Commands

This protocol does not define any commands.

6.4.2.2 Return indications

This protocol does not return any indications.

6.4.2.3 Local common data

This protocol does not define any local common data.

6.4.2.4 Public data

This protocol shall make the following data public:

- All data defined as Static Attribute, Static Non-Attribute Data, and local common data.

6.4.3 Protocol data unit

The protocol data unit for this protocol is a ROHC Packet. This protocol does not add any header.

6.4.4 Procedures and messages for the InConfiguration instance of the protocol

6.4.4.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

6.4.4.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

6.4.4.3 Message formats

This protocol does not define any messages.

6.4.5 Procedures and messages for the InUse instance of the protocol

6.4.5.1 Procedures

6.4.5.1.1 Protocol initialization

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in 15.1.2.

6.4.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

6.4.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

6.4.5.1.4 ROHC Support Protocol header

The ROHC Support Protocol does not add any header.

6.4.5.1.5 Command processing

This protocol does not define any commands.

6.4.5.1.6 Common procedures

Access network and access terminal send and receive packets as defined in IETF RFC 3095.

6.4.5.1.6.1 Use of multiple ROHC versions

The ROHC profile is a 16-bit integer. The eight LSBs distinguish the profile (e.g., RTP, UDP) and the eight MSBs distinguish the ROHC version. In ROHC packets, the profile identifier is abbreviated as an 8-bit field, carrying only the eight LSBs. Thus, if two or more profiles with the same eight LSBs are supported, ROHC needs a way to uniquely identify the version of the profile being used.

The access terminal shall use the 16-bit representation of the profiles carried in the Profiles field of ROHCParametersFwd to identify the ROHC version to which the 8-bit profile identifier carried in the ROHC packet refers. The access network shall use the 16-bit representation of the profiles carried in the Profiles field of ROHCParametersRev to identify the ROHC version to which the 8-bit profile identifier carried in the ROHC packet refers.

6.4.5.1.6.2 Type definitions

Protocol Type values assigned to protocols are specified in Clause 16. Protocol Type value for InUse and InConfiguration instance of this protocol is negotiated by Radio Link Protocol.

6.4.5.2 Message formats

This protocol does not define any messages.

6.4.6 Interface to other protocols

6.4.6.1 Commands sent

This protocol does not issue any commands.

6.4.6.2 Indications

This protocol does not register to receive any indications.

6.4.7 Configuration attributes

6.4.7.1 Simple attributes

This protocol does not define any simple attributes.

6.4.7.2 Complex attributes

This protocol defines following complex attributes.

6.4.7.2.1 ROHCPParametersFwd attribute

The sender shall set AttributeID field to 0x8000 (see Table 7).

Table 7—ROHCPParametersFwd attribute

Field	Length (bits)	Default
MaxCID	16	0
LargeCIDs	1	0
FeedbackForIncluded	1	N/A
FeedbackFor	5 ^a	N/A
MRRU	16	0
ProfileCount	8	4

ProfileCount occurrences of the following field:

Profile	16	0x0000 (uncompressed), 0x0001 (RTP), 0x0002 (UDP), 0x0003 (ESP)

Padding	0–7 (as needed)	N/A

^aThis field may be absent (see field descriptions for conditions).

MaxCID

The sender shall set this field to the MAX_CID parameter (see IETF RFC 3095) for this ROHC Channel. The sender shall not set this field to a value greater than MaxSupportedMaxCID.

LargeCIDs	If the LARGE_CIDs parameter (see IETF RFC 3095) for this ROHC Channel is false, then the sender shall set this field to '0'. Otherwise, the sender shall set this field to '1'. The sender shall not set this field to '1' if LargeCIDSupported is not set to '1'.
FeedbackForIncluded	If ROHC feedback associated with another Stream (ROHC channel) is sent on this Stream (ROHC channel), then this field shall be set to '1'. Otherwise, this field shall be set to '0'.
FeedbackFor	If FeedbackForIncluded is set to '0', then the sender shall omit this field. Otherwise, the sender shall set this field to the Stream number (ROHC channel) to which ROHC feedback sent on this Stream (ROHC channel) refers.
MRRU	The sender shall set this field to the MRRU parameter for this ROHC channel. The sender shall not set this field to a value larger than MaxSupportedMRRU.
ProfileCount	The sender shall set this field to the number of ROHC profiles supported by the decompressor.
Profile	The sender shall set this field to the ROHC profile supported by the decompressor according to the IANA ROHC Profile Identifiers. The sender shall not set this field to a value that is not included in the list of supported Profiles. The sender shall not set two or more Profile values with same value of eight LSBs.
Padding	The sender shall add Padding bits to make the length of the entire record an integer number of octets. The sender shall set these bits to '0'. The receiver shall ignore this field.

Commit	Soft

Scope	Dynamic

6.4.7.2.2 ROHCPParametersRev attribute

The sender shall set AttributeID field to 0x8001. ProfileCount occurrences of the following field (see Table 8).

Table 8—ROHCPParametersRev attribute

Field	Length (bits)	Default
MaxCID	16	0
LargeCIDs	1	0
FeedbackForIncluded	1	N/A
FeedbackFor	5 ^a	N/A
MRRU	16	0
TimerBasedCompression	1	1
ProfileCount	8	4

Table 8—ROHCParametersRev attribute (continued)

Field	Length (bits)	Default
ProfileCount occurrences of the following field:		
Profile	16	0x0000 (uncompressed), 0x0001 (RTP), 0x0002 (UDP), 0x0003 (ESP)
Padding	0–7 (as needed)	N/A

^aThis field may be absent (see field descriptions for conditions).

MaxCID	The sender shall set this field to the MAX_CID parameter (see IETF RFC 3095) for this ROHC Channel. The sender shall not set this field to a value greater than MaxSupportedMaxCID.
LargeCIDs	If the LARGE_CIDS parameter (see IETF RFC 3095) for this ROHC Channel is false, then the sender shall set this field to '0'. Otherwise, the sender shall set this field to '1'. If LargeCIDSupported is '0', then the sender shall not set this field to '1'.
FeedbackForIncluded	If ROHC feedback associated with another Stream (ROHC channel) is sent on this Stream (ROHC channel), then this field shall be set to '1'. Otherwise, this field shall be set to '0'.
FeedbackFor	If FeedbackForIncluded is set to '0', then the sender shall omit this field. Otherwise, the sender shall set this field to the Stream number (ROHC channel) to which ROHC feedback sent on this Stream (ROHC channel) refers.
MRRU	The sender shall set this field to the MRRU parameter for this ROHC channel. The sender shall not set this field to a value greater than MaxSupportedMRRU.
TimerBasedCompression	The sender shall set this field to '0' if timer-based compression according to IETF RFC 3095 is not enabled for this ROHC channel. The sender shall set this field to '1' if timer-based compression according to IETF RFC 3095 is enabled for this ROHC channel. If TimerBasedCompressionSupported is set to '0', then the sender shall not set this field to '1'.
ProfileCount	The sender shall set this field to the number of ROHC profiles supported by the decompressor.
Profile	The sender shall set this field to the ROHC profile supported by the decompressor according to the IANA ROHC Profile Identifiers. ¹¹ The sender shall not set this field to a value that is not included in the list of supported profiles. The sender shall not set two or more Profile values with same value of eight. LSBs.

¹¹IANA ROHC Profile Identifiers can be found at <http://www.iana.org/assignments/rohc-pro-ids>.

Padding The sender shall add Padding bits to make the length of the entire record an integer number of octets. The sender shall set these bits to '0'. The receiver shall ignore this field.

Commit	Soft

Scope	Dynamic

6.4.8 Non-attribute data

This protocol does not define any non-attribute data.

6.4.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records. All configuration attributes and non-attribute data are Session State Information records. This protocol does not define additional parameter records.

6.5 Basic EAP Support Protocol

6.5.1 Overview

The EAP Support Protocol performs the following functions:

- Sends and receives EAP protocol packets
- Negotiates if the access terminal would initiate EAP Fast Reauthentication when Route Open is initiated
- Provides control messages for EAP

6.5.2 Primitives, local common data, and public data

6.5.2.1 Commands

This protocol does not define any commands.

6.5.2.2 Return indications

This protocol does not return any indications.

6.5.2.3 Local common data

This protocol does not define any local common data.

6.5.2.4 Public data

This protocol shall make the following data public:

- All data defined as Static Attribute, Static Non-Attribute Data, and local common data.

6.5.3 Protocol data unit

The protocol data unit for this protocol is an EAP Packet. This protocol does not add any header.

6.5.4 Procedures and messages for the InConfiguration instance of the protocol

6.5.4.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

6.5.4.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

6.5.4.3 Message formats

This protocol does not define any messages.

6.5.5 Procedures and messages for the InUse instance of the protocol

6.5.5.1 Procedures

6.5.5.1.1 Protocol initialization

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in 15.1.2.

6.5.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

6.5.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

6.5.5.1.4 EAP Support Protocol header

The EAP Support Protocol does not add any header.

6.5.5.1.5 Command processing

This protocol does not define any commands.

6.5.5.1.6 Common procedures

Access network and access terminal send and receive EAP packets as defined in IETF RFC 3748.

6.5.5.1.7 Access terminal procedures

The access terminal shall initiate EAP Fast Reauthentication procedure when Route Open is initiated, if and only if the following condition is true:

- InitiateEAPFastReauthentication attribute of this protocol in the Session Anchor Route is set to 0x000000.

The access terminal may send InitiateEAP message to request access network to initiate EAP procedure as specified in IETF RFC 3748.

If access terminal receives InitiateEAPFastReauthentication message and the EAPFastReauthenticationSupported field of ATSupportedApplicationProtocolIDs public data of the Session Control Protocol is set to '1', the access terminal shall perform one of the following:

- Initiate EAP Fast Reauthentication procedure, if authentication has been previously established to this AN.
- Send InitiateEAP message, if no authentication for this AN is in effect.

6.5.5.1.8 Access network procedures

The access network may send InitiateEAPFastReauthentication message to request access terminal to initiate EAP Fast Reauthentication procedure.

If access network receives InitiateEAP message, the access network shall Initiate EAP procedure as specified in IETF RFC 3748.

6.5.5.2 Type definitions

Protocol Type values assigned to protocols are specified in 16.1. Protocol Type value for InUse and InConfiguration instance of this protocol is negotiated by Radio Link Protocol.

6.5.5.3 Message formats

6.5.5.3.1 InitiateEAPFastReauthentication

The access network sends this message to request the access terminal to initiate EAP Fast Reauthentication procedures (see Table 9).

Table 9—InitiateEAPFastReauthentication

Field	Length (bits)
MessageID	8

MessageID The sender shall set this field to 0x00.

Channels	FTC RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

6.5.5.3.2 InitiateEAP

The access terminal sends this message to request the access network to initiate EAP procedures (see Table 10).

Table 10—InitiateEAP

Field	Length (bits)
MessageID	8

MessageID The sender shall set this field to 0x01.

Channels	FTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

6.5.6 Interface to other protocols

6.5.6.1 Commands sent

This protocol does not issue any commands.

6.5.6.2 Indications

This protocol does not register to receive any indications.

6.5.7 Configuration attributes

6.5.7.1 Simple attributes

The negotiable simple attributes for this protocol is listed in Table 11. The access terminal and the access network shall use as defaults the values in Table 11 typed in ***bold italics***.

Table 11—Configurable values

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0000	InitiateEAPFastReauthentication	Soft/ Dynamic	0x000000	Access terminal to initiate EAP Fast Reauthentication when Route Open is initiated.
			<i>0x111111</i>	Access terminal to not initiate EAP Fast Reauthentication when Route Open is initiated.
			All other values	Reserved

6.5.7.2 Complex attributes

This protocol does not define any complex attributes.

6.5.8 Non-attribute data

This protocol does not define any non-attribute data.

6.5.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records. All configuration attributes and non-attribute data are Session State Information records. This protocol does not define additional parameter records.

7. Radio Link Sublayer

7.1 Introduction

7.1.1 General overview

The Radio Link Sublayer provides the following functions:

- a) Negotiation of packet filters and Quality of Service (QoS) for IP flows.
- b) Mapping of Reservations to Streams.
- c) Segmentation, reassembly, retransmission (if needed) and duplicate detection of higher sublayer packets.
- d) Multiplexing of application streams. Streams listed below are reserved. The other streams can be assigned to applications with different QoS requirements, or other applications.
 - Stream 0: Broadcast Signaling on forward link and Reserved on reverse link
 - Stream 1: Best Effort Delivery Signaling
 - Stream 2: Reliable Delivery Signaling
 - Stream 3: Broadcast Inter Route Tunneling on forward link and Reserved on reverse link
 - Stream 4: Best Effort Delivery Inter Route Tunneling
 - Stream 5: Best Effort Delivery Inter Route Tunneling
 - Stream 6: Best Effort Delivery Inter Route Tunneling
 - Stream 31: Reserved for future use
- e) Identify the Protocol Stack associated with this route during route initialization.
- f) Determine whether Route Protocol packets for transmission should be delivered to the Packet Consolidation Protocol of this Route or the Inter-Route Tunneling Protocol of this or another route.

The Radio Link Sublayer uses one instance of the Radio Link Protocol for each stream. A single overall instance of the QoS Management Protocol, the Stream Protocol, and the Route Protocol are instantiated to implement the Radio Link Sublayer.

Application subtypes are specified in 16.1.

7.1.2 Data encapsulation for the Radio Link Sublayer

Figure 20 illustrates the relationship between an Application Sublayer packet, a Radio Link Protocol packet, a Stream Protocol packet, a Route Protocol packet, and a Lower MAC Sublayer payload.

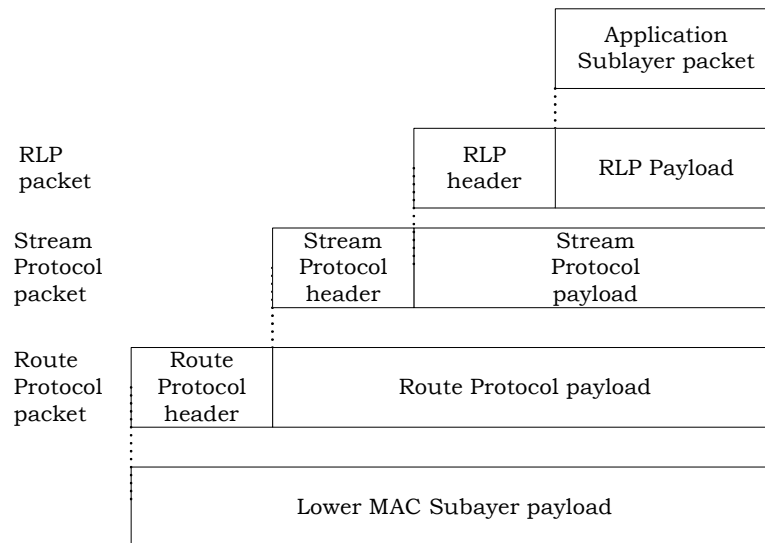


Figure 20—Radio Link Sublayer encapsulation

7.2 Basic QoS Management Protocol

7.2.1 Introduction

7.2.1.1 General overview

The Basic QoS Management Protocol provides the following:

- Negotiation of packet filters and Quality of Service (QoS) for IP packets.
- Mapping of Reservations to Streams.
- Flow control for each reservation.

The protocol subtype for this protocol is defined in 16.1. This protocol uses the Basic Signaling Protocol to transmit and receive messages.

The QoS Management Protocol implements flow control techniques to manage data transmission from the AN to the AT. The flow control state is tracked at the AN through the variable XonStatus, for each Forward Flow Reservation. XonStatus can have the values “ON” or “OFF.” When XonStatus is set to “ON” the AN can send data to the AT; conversely when XonStatus is set to “OFF” the AN does not send data to the AT. There is no flow control mechanism to prevent the AT from sending data to the AN.

7.2.1.2 Primitives and public data

7.2.1.2.1 Commands

This protocol does not define any commands.

7.2.1.2.2 Return indications

This protocol does not return any indications.

7.2.1.2.3 Local common data

This protocol does not define any local common data.

7.2.1.2.4 Public data

- Subtype for this protocol.
- All data defined as Static Attribute, Static Non-Attribute Data, and local common data.

7.2.1.3 Data encapsulation for the InUse instance of the protocol

This protocol does not add any headers or trailers. This protocol filters higher layer IP packets and forwards them to the appropriate Stream.

7.2.2 Procedures and messages for the InConfiguration instance of the protocol

7.2.2.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

7.2.2.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

The access terminal shall not initiate negotiation of the ANSupportedQoSProfiles attribute. The access network shall not initiate negotiation of the ATSupportedQoSProfiles attribute. The access terminal shall ensure that ATSupportedQoSProfiles attribute includes all supported values of ProfileValue with ProfileType equal to 0x04 by initiating negotiation of ATSupportedQoSProfiles attribute if necessary. The access network shall ensure that ANSupportedQoSProfiles attribute includes all supported values of ProfileValue with ProfileType equal to 0x04 by initiating negotiation of ANSupportedQoSProfiles attribute if necessary.

If NetworkInitiatedQoSAllowed is 0x00, then the access network shall not initiate negotiation of the ReservationKKQoSRequestFwd, ReservationKKQoSRequestRev, ReservationKKPacketFiltersFwd, and ReservationKKPacketFiltersRev attributes.

The access network shall accept a request from the access terminal to set the following combination of attributes to default value for same value of KK, if the request is in a single message:

- ReservationKKQoSRequestFwd and ReservationKKStreamFwd
- ReservationKKQoSRequestFwd
- ReservationKKQoSRequestRev and ReservationKKStreamRev
- ReservationKKQoSRequestRev

If ReservationKKQoSRequestFwd is set to the default value, the access network shall accept the request from the access terminal to set the ReservationKKStreamFwd attribute to the default value for the same value of KK.

If ReservationKKQoSRequestRev is set to the default value, the access network shall accept the request from the access terminal to set ReservationKKStreamRev attribute to the default value for the same value of KK.

7.2.2.3 Message formats

This protocol does not define any messages.

7.2.3 Procedures and messages for the InUse instance of the protocol

7.2.3.1 7.2.3.1 Reservation definition

The packet filter and requested QoS associated with an IP flow is called a Reservation. Each Reservation is identified by the direction of the Reservation (forward or reverse) and a ReservationLabel, which is a two-digit hexadecimal number in the range 0x00 to 0xfe, inclusive.

XonStatus associated with each Reservation indicates if the access network is allowed to transmit packets belonging to the higher layer flow identified by that Reservation.

7.2.3.2 Procedures

7.2.3.2.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and the access network shall perform the following in the order specified:

- Perform the procedures specified in 15.1.2.
- The access network shall set the Forward Reservation with ReservationLabel 0xfe to the Open state.¹²
- The access terminal shall set the Reverse Reservation with ReservationLabel 0xfe to the Open state.
- The access network shall set the Forward Reservations with ReservationLabel not equal to 0xfe to the Close state.
- The access terminal shall set the Reverse Reservations with ReservationLabel not equal to 0xfe to the Close state.
- The access network shall set XonStatus to ON for all the Forward Reservations.

7.2.3.2.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

7.2.3.2.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

7.2.3.2.4 Reservation procedures

7.2.3.2.4.1 Reservation State Maintenance

The Stream parameter of the ReservationKKStreamFwd or ReservationKKStreamRev attribute indicates the Stream associated with ReservationLabel KK. Each forward ReservationLabel shall be associated with no more than one forward Stream. Each reverse ReservationLabel shall be associated with no more than one reverse Stream.

¹²Forward and reverse link Reservations 0xfe are initialized in the Open state so that data can be sent without having to perform a state transition.

Each Reservation can be in one of the following two states:

- Close State
- Open State

All the Forward Link Reservations for which the ProfileType field of ReservationKKQoSRequestFwd attribute is set to 0x00 shall be in Close State. State transitions specified in this subclause apply only to Forward Link Reservations for which the ProfileType field of ReservationKKQoSRequestFwd attribute is not set to 0x00.

All the Reverse Link Reservations for which the ProfileType field of ReservationKKQoSRequestRev attribute is set to 0x00 shall be in Close State. State transitions specified in this subclause apply only to Reverse Link Reservations for which the ProfileType field of ReservationKKQoSRequestRev attribute is not set to 0x00.

If the transmitter is the access network and XonStatus for a Forward Reservation is set to OFF, then the access network shall not transmit packets belonging to the higher layer flow identified by that Reservation. Otherwise, the following transmitter requirements apply:

- If a Reservation is in the Open state and if the Stream associated with the Reservation is activated, then the transmitter should transmit higher-layer packets using the Stream associated with the Reservation.
- The transmitter shall not transmit higher-layer packets using any Stream other than the Stream associated with the Reservation or the Stream bound to ReservationLabel 0xfe.
- The transmitter should transmit higher-layer packets belonging to a higher layer flow that is not associated with any Stream using the Stream bound to ReservationLabel 0xfe.
- The transmitter may transmit higher-layer packets belonging to a higher layer flow identified by a Reservation that is in the Close state using the Stream bound to ReservationLabel 0xfe.
- The transmitter may transmit higher-layer packets belonging to a higher layer flow identified by a Reservation that is bound to a de-activated Stream using the Stream bound to ReservationLabel 0xfe.

The access terminal and the access network shall not perform any actions that will result in the number of Reverse Link Reservations in the Open state to exceed the value specified by MaxNumOpenReservationsRev. The access terminal and the access network shall not perform any actions that will result in the number of Forward Link Reservations in the Open state to exceed the value specified by MaxNumOpenReservationsFwd.

Figure 21 and Figure 22 show the state transition diagram at the access terminal and the access network.

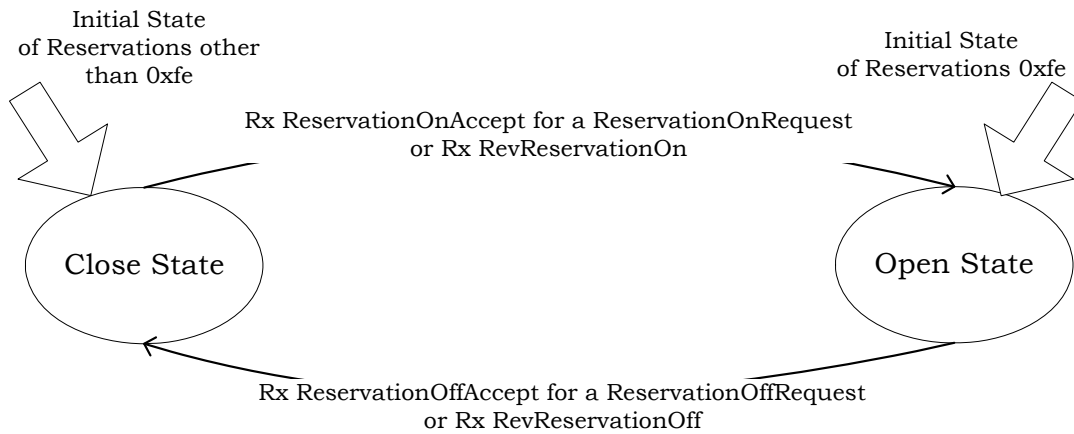


Figure 21—Reverse Link Reservation state diagram (access terminal)

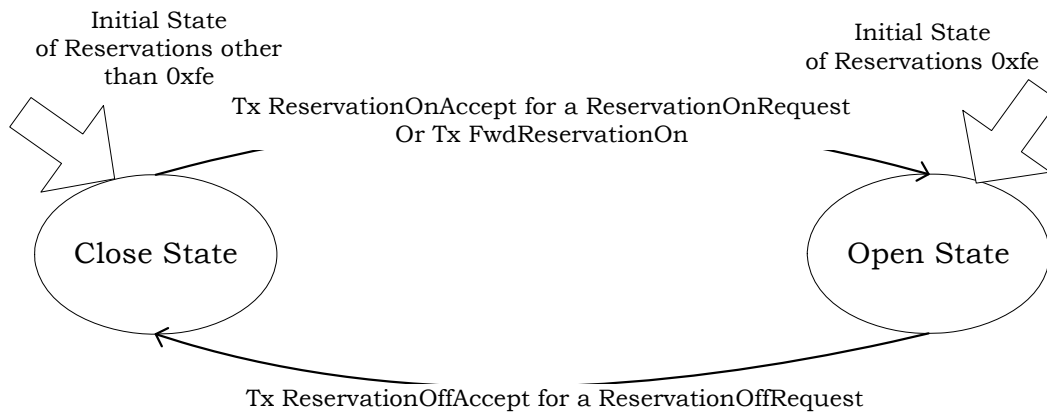


Figure 22—Forward Link Reservation state diagram (access network)

7.2.3.2.4.1.1 State-independent requirements

7.2.3.2.4.1.1.1 Packet Filter processing requirements

In the following requirements, ReservationKKPacketFilters refers to ReservationKKPacketFiltersFwd if the sender is an access network and refers to ReservationKKPacketFiltersRev if the sender is an access terminal. When an IP packet is to be transmitted, the sender shall conform to following requirements:

- a) The sender shall search for a match among all packet filters specified by all ReservationKKPacketFilters attributes for which NumPacketFilters is not set to 0x00. A packet filter is considered to be a match if all the PacketFilterComponent match with the IP packet to be transmitted.
- b) If multiple packet filters match, then:
 - If all of the packet filters have EvaluationPrecedence value of 0xff, then sender shall consider the IP packet to be matched with one of the packet filters based on implementation-dependent algorithm.

- Otherwise, sender shall consider the IP packet to be matched with the packet filter with lowest value of EvaluationPrecedence.
- c) The sender shall transmit the IP packet using a stream as per the requirements in 7.2.3.2.4.1 for the ReservationLabel corresponding to the matched packet filter. If none of the packet filters match, then the sender should transmit the IP packet using the Stream bound to ReservationLabel 0xfe.

7.2.3.2.4.1.1.2 Access terminal requirements

Upon receiving a RevReservationOn message, the access terminal shall

- Respond with a RevReservationAccept message within the time period specified by $T_{QMPResponse}$ of receiving the RevReservationOn message.
- Set the TransactionID field of the RevReservationAccept message to that of the RevReservationOn message.

Upon receiving a RevReservationOff message, the access terminal shall

- Respond with a RevReservationAccept message within the time period specified by $T_{QMPResponse}$ of receiving the RevReservationOff message.
- Set the TransactionID field of the RevReservationAccept message to that of the RevReservationOff message.

Upon receiving a FwdReservationOn message, the access terminal shall

- Respond with a FwdReservationAck message within the time period specified by $T_{QMPResponse}$ of reception of the FwdReservationOn message.
- Set the TransactionID field of the FwdReservationAck message to that of the FwdReservationOn message.

Upon receiving a FwdReservationOff message, the access terminal shall

- Respond with a FwdReservationAck message within the time period specified by $T_{QMPResponse}$ of receiving the FwdReservationOff message.
- Set the TransactionID field of the FwdReservationAck message to that of the FwdReservationOff message.

Upon receiving a ReservationReject message in response to ReservationOnRequest or ReservationOffRequest message

- The access terminal shall start a Retry Delay timer for time specified by RetryDelay field in the ReservationReject message.
- While the Retry Delay timer is running, the access terminal should not request state transition requested in the ReservationOnRequest or ReservationOffRequest message.

Upon sending an XonRequest message, the access terminal shall

- Not resend an XonRequest message until $T_{QMPResponse}$ time units have elapsed with no corresponding XonResponse message.

Upon sending an XoffRequest message, the access terminal shall

- Not resend an XoffRequest message until $T_{QMPResponse}$ time units have elapsed with no corresponding XoffResponse message.

The access terminal shall not include a Forward Reservation in ReservationOnRequest message if corresponding ReservationKKRoRAAllowedFwd attribute is set to 0x00. The access terminal shall not

include a Reverse Reservation in ReservationOnRequest message if corresponding ReservationKKRoRAAllowedRev attribute is set to 0x00.

7.2.3.2.4.1.1.3 Access network requirements

The access network may send a RevReservationOn message to transition the state of the Reverse Link Reservation of the access terminal to the Open state.

The access network may send a RevReservationOff message to transition the state of the Reverse Link Reservation of the access terminal to the Close state.

If the access network receives a ReservationOnRequest message, it shall

- a) Send either a ReservationOnAccept message or a ReservationReject message within the time period specified by $T_{QMPResponse}$ of reception of the ReservationOnRequest message.
- b) Set the TransactionID field of the ReservationOnAccept or ReservationReject message to that of the ReservationOnRequest message.
- c) If ReservationOnAccept message is sent, then in the ReservationOnAccept message the access network shall include all the Reservations requested in ReservationOnRequest message. The access network shall set QoS_ATTRIBUTE_SET_ID field of each accepted Reservation as specified below:
 - If QoSAttributeSetIDCount field of the ReservationOnRequest message was set to value other than 0x0 for this Reservation, set the QoS_ATTRIBUTE_SET_ID field of the accepted Reservation in ReservationOnAccept message to the value of any QoSAttributeSetID field of the Reservation in the ReservationOnRequest message.
 - Otherwise, set the QoS_ATTRIBUTE_SET_ID field of the accepted Reservation in ReservationOnAccept message to the value of any QoS_ATTRIBUTE_SET_ID field in the corresponding ReservationKKQoSRequestFwd (for Forward Link Reservation) or ReservationKKQoSRequestRev (for Reverse Link Reservation) attribute.

Before sending a ReservationOnAccept, a RevReservationOn, or a FwdReservationOn message, the access network should make sure that all the QoS related attributes are configured properly to achieve the granted QoS.

If the access network receives a ReservationOffRequest message, it shall

- Send a ReservationOffAccept or a ReservationReject message within the time period specified by $T_{QMPResponse}$ of reception of the ReservationOffRequest message.
- Set the TransactionID field of the ReservationOffAccept or ReservationReject message to that of the ReservationOffRequest message.

If the access network receives an XonRequest message, it shall

- If ReservationCount field of the XonRequest message is set to 0x00, then set XonStatus to ON for all the Forward Reservations. Otherwise, set XonStatus to ON for all the Forward Reservations specified in the XonRequest message.
- Send an XonResponse message within the time period specified by $T_{QMPResponse}$ of reception of the XonRequest message.
- Set the TransactionID field of the XonResponse message to that of the XonRequest message.

If the access network receives an XoffRequest message, it shall

- If ReservationCount field of the XoffRequest message is set to 0x00, then set XonStatus to OFF for all the Forward Reservations. Otherwise, set XonStatus to OFF for all the Forward Reservations specified in the XoffRequest message.

- Send an XoffResponse message within the time period specified by $T_{QMPResponse}$ of reception of the XoffRequest message.
- Set the TransactionID field of the XoffResponse message to that of the XoffRequest message.

The access network shall not transmit higher-layer packets belonging to a higher layer flow identified by a Forward Reservation if XonStatus for that Reservation is set to OFF.

7.2.3.2.4.2 Close state

7.2.3.2.4.2.1 Access terminal requirements

The access terminal shall not transmit PDUs from higher layer flows belonging to this Reservation using any Stream other than the Stream associated with ReservationLabel 0xfe. The access terminal may send a ReservationOnRequest message to request transition of the Reservation to the Open state.¹³ The access terminal may resend a ReservationOnRequest message if it does not receive a corresponding ReservationOnAccept or ReservationReject message within the time period specified by $T_{QMPResponse}$ of sending the ReservationOnRequest message. If the ReservationOnRequest message contains a Reverse Reservation, then the Reservation shall transition to the Open state when the access terminal receives the corresponding ReservationOnAccept message.

Upon receiving a RevReservationOn message, the access terminal shall transition the Reservation to the Open state.

7.2.3.2.4.2.2 Access network requirements

If the Reservation entered this state as a result of any condition other than the following conditions, then the access network shall send a FwdReservationOff message upon entering this state:

- The access network transmitted a ReservationOffAccept message in response to a ReservationOffRequest message requesting to transition the Reservation to the Close state.
- The Reservation transitioned to the Close state because the access terminal transitioned to the Idle state.

Upon sending a ReservationOnAccept message for a Forward Reservation in response to a ReservationOnRequest message, the access network shall transition the Reservation to the Open state.

Upon sending a FwdReservationOn message, the access network shall transition the Reservation to the Open state.

7.2.3.2.4.3 Open state

7.2.3.2.4.3.1 Access terminal requirements

The access terminal may transmit PDUs from higher layer flows belonging to this Reservation using the Stream to which the Reservation is bound.

The access terminal may send a ReservationOffRequest message to request the transition of a Reservation to the Close state. The access terminal may resend a ReservationOffRequest message if it does not receive a ReservationOffAccept or ReservationReject message within the time period specified by $T_{QMPResponse}$ of sending the ReservationOffRequest message. If the ReservationOffRequest message contains a Reverse

¹³Note that the ReservationOnRequest message supports requests for multiple Reservations on both the forward and reverse links. This arrangement allows requests for groups of Reservations (e.g., for bidirectional higher layer application flows) to be combined in the same ReservationOnRequest message.

Reservation, then the access terminal shall transition the Reservation to the Close state when the access terminal receives a ReservationOffAccept message.

Upon receiving a RevReservationOff message, the access terminal shall transition the Reservation to the Close state. Upon receiving an *AirLinkManagement.IdleInitiated* indication, the access terminal shall transition all the Reservations to the Close state, except the Reservation with ReservationLabel 0xfe.

The access terminal may send a ReservationOnRequest message to request the transition of a Reservation to the Open state. The access terminal may resend a ReservationOnRequest message if it does not receive a ReservationOnAccept or ReservationReject message within the time period specified by $T_{QMPResponse}$ of sending the ReservationOffRequest message. If the ReservationOnRequest message contains a Reverse Reservation, then the access terminal shall transition the Reservation to the Close state when the access terminal receives a ReservationReject message.

7.2.3.2.4.3.2 Access network requirements

If XonStatus for this Forward Reservation is set to ON, then the access network may transmit PDUs from higher layer flows belonging to this Reservation using the Stream to which the Reservation is bound.

Upon sending a ReservationOffAccept message for a Forward Reservation in response to a ReservationOffRequest message, the access network shall transition the Reservation to the Close state.

Upon receiving an *AirLinkManagement.IdleInitiated* indication, the access network shall transition to the Close state all the Reservations, except the Reservation with ReservationLabel 0xfe.

Upon sending a ReservationReject message for a Forward Reservation in response to a ReservationOnRequest message, the access network shall transition the Reservation to the Close state.

7.2.3.3 Message formats

The messages described in this subclause control the function of the Basic QoS Management Protocol. These messages are exchanged between the access terminal and the access network using the Signaling Application.

Throughout this subclause and the next, a ProfileType field is used by both messages and attributes. The allowable values for ProfileType are described in Table 12.

Table 12—ProfileType encoding

ProfileType	Meaning	Allowed in QoS Requests	Allowed in QoS Grants	Allowed in attributes specifying supported QoS profiles
0x00	NULL	Yes	No	No
0x01	Requested QoS	Yes	No	No
0x02	Requested QoS ^a	Yes	No	No
0x03	Granted QoS	No	Yes	No
0x04	FlowProfileID as defined in 3GPP2 C.R1001-F v1.0	No	No	Yes
All other values	Reserved	N/A	N/A	N/A

^aTwo values of ProfileType are defined for Requested QoS to maintain compatibility with ProfileType values used in High Rate Packet Data system.

7.2.3.3.1 ReservationOnRequest

The access terminal sends this message to request transition of one or more Reservations to the Open State (see Table 13).

Table 13—ReservationOnRequest message

Field	Length (bits)
MessageID	8
TransactionID	8
EmergencyIndication	1
ReservationCount	7

ReservationCount occurrences of the following record:

ReservationLabel	8
Link	1
Reserved1	3
QoSAttributeSetIDCount	4

QoSAttributeSetIDCount occurrences of the following two fields:

QoSAttributeSetID	7
Reserved2	1

MessageID	The access terminal shall set this field to 0x00.
TransactionID	The access terminal shall set this field to one more (modulo 256) than the TransactionID field of the last ReservationOnRequest, or ReservationOffRequest message sent by the access terminal. If this is the first ReservationOnRequest, or ReservationOffRequest message sent by the access terminal, then the access terminal shall set this field to zero.
EmergencyIndication	If this is an emergency ReservationOnRequest, then the access terminal shall set this field to '1'. Otherwise, the access terminal shall set this field to '0'.
ReservationCount	The access terminal shall set this field to the number of the following records in this message.
ReservationLabel	The access terminal shall set this field to the ReservationLabel for which this request is generated. The access terminal shall not set this field to a value greater than (MaxNumOpenReservationsFwd + 1 + MaxNumOpenReservationsRev + 1), where MaxNumOpenReservationsFwd and MaxNumOpenReservationsRev are fields of the MaxReservations attribute.
Link	If this request is for a Forward Reservation, then the access terminal shall set this field to '1'. If this request is for a Reverse Reservation, then the access terminal shall set this field to '0'.
Reserved1	The access network shall ignore this field.

QoSAttributeSetIDCount The access terminal shall set this field to the number of QoSAttributeSetID fields in this message. The access terminal shall include QoSAttributeSetID field in descending order of preference. The access terminal may set this field to 0x0 to request QoS_ATTRIBUTE_SET_ID preference ordering as specified in the corresponding ReservationKKQoSRequestFwd or ReservationKKQoSRequestRev attribute. The access terminal shall include QoSAttributeSetIDCount occurrences of the following two fields with the message.

QoSAttributeSetID The access terminal shall set this field to the value of any QoS_ATTRIBUTE_SET_ID field in the corresponding ReservationKKQoSRequestFwd or ReservationKKQoSRequestRev attribute.

Reserved2 The access network shall ignore this field.

Channels	RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

7.2.3.3.2 ReservationOffRequest

The access terminal sends this message to request transition of one or more Reservations to the Close State (see Table 14).

Table 14—ReservationOffRequest

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following three fields:

ReservationLabel	8
Link	1
Reserved	7

MessageID The access terminal shall set this field to 0x01.

TransactionID The access terminal shall set this field to one more (modulo 256) than the TransactionID field of the last ReservationOnRequest, or ReservationOffRequest message sent by the access terminal. If this is the first ReservationOnRequest, or ReservationOffRequest message sent by the access terminal, then the access terminal shall set this field to zero.

ReservationCount The access terminal shall set this field to the number of the following three fields in this message. The access terminal shall not set this field to a value greater than (MaxNumOpenReservationsFwd + 1 + MaxNumOpenReservationsRev + 1), where MaxNumOpenReservationsFwd and MaxNumOpenReservationsRev are fields of the MaxReservations attribute.

ReservationLabel	The access terminal shall set this field to the Reservation for which this request is generated.
Link	If this request is for a Forward Reservation, then the access terminal shall set this field to '1'. If this request is for a Reverse Reservation, then the access terminal shall set this field to '0'.
Reserved	The access network shall ignore this field.

Channels	RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required

7.2.3.3.3 ReservationOnAccept

The access network sends this message to acknowledge reception of and allow the state transition requested by a ReservationOnRequest message (see Table 15).

Table 15—ReservationOnAccept message

Field	Length (bits)
MessageID	8
TransactionID	8
ReservationCount	8

ReservationCount occurrences of the following six fields:

ReservationLabel	8
Link	1
Reserved	7
ProfileType	8
ProfileLength	16
ProfileValue	ProfileLength × 8

MessageID	The sender shall set this field to 0x02.
TransactionID	The access network shall set this field to the TransactionID field of the ReservationOnRequest message to which the access network is responding.
ReservationCount	The access network shall set this field to the number of the following six fields in this message. The access network shall not set this field to a value greater than (MaxNumOpenReservationsFwd + 1 + MaxNumOpenReservationsRev + 1), where MaxNumOpenReservationsFwd and MaxNumOpenReservationsRev are fields of the MaxReservations attribute.
ReservationLabel	The access network shall set this field to the ReservationLabel for which this accept is generated.

Link	If this accept is for a Forward Reservation, then the access network shall set this field to '1'. If this accept is for a Reverse Reservation, then the access network shall set this field to '0'.
Reserved	The access terminal shall ignore this field.
ProfileType	The access network shall set this field to indicate the profile type corresponding to granted QoS for this ReservationLabel. The access network shall set this field to 0x03. See Table 12 for encoding of ProfileType field.
ProfileLength	The access network shall set this field to length of the ProfileValue field in units of octets.
ProfileValue	The access network shall set this field to the Profile Value corresponding to the Profile Type according to 7.2.5.2.1.1.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required when key is available

7.2.3.3.4 ReservationOffAccept

The access network sends this message to acknowledge reception of and allow the state transition requested by a ReservationOffRequest message (see Table 16).

Table 16—ReservationOffAccept message

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The sender shall set this field to 0x03.

TransactionID The access network shall set this field to the TransactionID field of the ReservationOffRequest message to which the access network is responding.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required

7.2.3.3.5 ReservationReject

The access network sends this message to acknowledge reception of and deny the state transition requested by a ReservationOnRequest or ReservationOffRequest message (see Table 17).

Table 17—ReservationReject message

Field	Length (bits)
MessageID	8
TransactionID	8
RejectReason	8
RetryDelay	8

MessageID	The access network shall set this field to 0x04.
TransactionID	The access network shall set this field to the TransactionID field of the ReservationOnRequest or ReservationOffRequest message to which the access network is responding.
RejectReason	The access network shall set this field to the reason for rejection as specified in Table 18.

Table 18—RejectReason encoding

RejectReason	Meaning
0x00	Network busy
0x01	Access terminal not authorized
0x02	Request queued
All other values	Reserved

RetryDelay	If the access network is sending this message in response to a ReservationOffRequest message, then the access network shall set this field to 0x00. The access network shall set this field as follows: Let $a = \text{RetryDelay}[4:0]$ and $b = \text{RetryDelay}[7:5]$. The effective retry delay is then computed as $2^b \times a$ in units of seconds.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required

7.2.3.3.6 RevReservationOn

The access network sends this message to transition a Reverse Reservation to the Open state (see Table 19).

Table 19—RevReservationOn message

Field	Length (bits)
MessageID	8
ReservationCount	8
ReservationCount occurrences of the following four fields:	
ReservationLabel	8

Table 19—RevReservationOn message (continued)

Field	Length (bits)
ProfileType	8
ProfileLength	16
ProfileValue	ProfileLength × 8

MessageID	The access network shall set this field to 0x05.
ReservationCount	The access network shall set this field to the number of the following four fields in this message. The access network shall not set this field to a value greater than (MaxNumOpenReservationsRev + 1), where MaxNumOpenReservationsRev is a field of the MaxReservations attribute.
ReservationLabel	The access network shall set this field to the Reservation that is to be transitioned to the Open state.
ProfileType	The access network shall set this field to indicate the profile type corresponding to granted QoS for this ReservationLabel. The access network shall set this field to 0x03. See Table 12 for encoding of ProfileType field.
ProfileLength	The access network shall set this field to length of the ProfileValue field in units of octets.
ProfileValue	The access network shall set this field to the Profile Value corresponding to the Profile Type according to 7.2.5.2.1.1.

Channels	FTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required

7.2.3.3.7 RevReservationOff

The access network sends this message to transition a Reverse Reservation to the Close state (see Table 20).

Table 20—RevReservationOff message

Field	Length (bits)
MessageID	8
Reason	8
ReservationCount	8
ReservationCount occurrences of the following field:	
ReservationLabel	8

MessageID	The access network shall set this field to 0x06.
Reason	The access network shall set this field to the reason for sending this message as specified in Table 21.

Table 21—Reason encoding

Reason	Meaning
0x00	Network busy
0x01	Access terminal not authorized
0x02	Normal Reservation close
All other values	Reserved

ReservationCount The access network shall set this field to the number of ReservationLabel fields in this message. The access network shall not set this field to a value greater than (MaxNumOpenReservationsRev + 1), where MaxNumOpenReservationsRev is a field of the MaxReservations attribute.

ReservationLabel The access network shall set this field to the Reservation that is to be transitioned to the Close state.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required

7.2.3.3.8 FwdReservationOff

The access network sends this message to inform the access terminal when a Forward Reservation transitions to the Close state (see Table 22).

Table 22—FwdReservationOff message

Field	Length (bits)
MessageID	8
Reason	8
ReservationCount	8

ReservationCount occurrences of the following field:

ReservationLabel	8

MessageID The access network shall set this field to 0x07.

Reason The access network shall set this field to the reason for sending this message as specified in Table 21.

ReservationCount The access network shall set this field to the number of ReservationLabel fields in this message. The access network shall not set this field to a value greater than (MaxNumOpenReservationsFwd + 1), where MaxNumOpenReservationsFwd is a field of the MaxReservations attribute.

ReservationLabel The access network shall set this field to the Reservation transitioned to the Close state.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required

7.2.3.3.9 FwdReservationOn

The access network sends this message to inform the access terminal when a Forward Reservation transitions to the Open state (see Table 23).

Table 23—FwdReservationOn message

Field	Length (bits)
MessageID	8
ReservationCount	8

ReservationCount occurrences of the following four fields:

ReservationLabel	8
ProfileType	8
ProfileLength	16
ProfileValue	ProfileLength × 8

MessageID	The access network shall set this field to 0x08.
ReservationCount	The access network shall set this field to the number of the occurrences of the following four fields in this message. The access network shall not set this field to a value greater than (MaxNumOpenReservationsFwd + 1), where MaxNumOpenReservationsFwd is a field of the MaxReservations attribute.
ReservationLabel	The access network shall set this field to the Reservation that transitioned to the Open state.
ProfileType	The access network shall set this field to indicate the profile type corresponding to granted QoS for this ReservationLabel. The access network shall set this field to 0x03. See Table 12 for encoding of ProfileType field.
ProfileLength	The access network shall set this field to length of the ProfileValue field in units of octets.
ProfileValue	The access network shall set this field to the Profile Value corresponding to the Profile Type according to 7.2.5.2.1.1.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required

7.2.3.3.10 XonRequest

The access terminal sends this message to allow access network to send data on one or more Forward Reservations (see Table 24).

Table 24—XonRequest message

Field	Length (bits)
MessageID	8
ReservationCount	8
ReservationCount occurrences of the following one field:	
ReservationLabel	8

MessageID	The access terminal shall set this field to 0x09.
ReservationCount	The access terminal shall set this field to the number of ReservationLabel fields in this message. The access terminal shall set this field to 0x00 to indicate that this request is for all Forward Reservations. The access terminal shall not set this field to a value greater than (MaxNumOpenReservationsFwd + 1), where MaxNumOpenReservationsFwd is a field of the MaxReservations attribute.
ReservationLabel	The access terminal shall set this field to the Forward ReservationLabel for which this request is generated.

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required

7.2.3.3.11 XoffRequest

The access terminal sends this message to request access network to stop transmission of data on one or more Forward Reservations (see Table 25).

Table 25—XoffRequest message

Field	Length (bits)
MessageID	8
ReservationCount	8
ReservationCount occurrences of the following one field:	
ReservationLabel	8

MessageID	The access terminal shall set this field to 0x0a.
ReservationCount	The access terminal shall set this field to the number of ReservationLabel fields in this message. The access terminal shall set this field to 0x00 to indicate that this request is for all Forward Reservations. The access terminal shall not set this field to a value greater than (MaxNumOpenReservationsFwd + 1), where MaxNumOpenReservationsFwd is a field of the MaxReservations attribute.

ReservationLabel The access terminal shall set this field to the Forward Reservation for which this request is generated.

Channels	RTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required

7.2.3.4 Interface to other protocols

7.2.3.4.1 Commands

This protocol does not issue any commands.

7.2.3.4.2 Indications

This protocol registers to receive the following indication:

- *AirLinkManagement.IdleInitiated*

7.2.4 Protocol numeric constants

Protocol numeric constants are listed in Table 26.

Table 26—Protocol numeric constants

Constant	Meaning	Value
$T_{QMPResponse}$	Time period within which the access network is to respond to ReservationOnRequest and ReservationOffRequest messages.	1 second

7.2.5 Configuration attributes for the Basic QoS Management Protocol

The number of Forward Link Reservations that satisfy at least one of the following conditions shall not exceed one plus the value of the MaxNumReservationsFwd field of MaxReservations attribute:

- The Reservation is in Open state.
- The Reservation is bound to a Stream.
- The ReservationKKQoSRequestFwd corresponding to the Reservation is set to a non-default value.

The number of Reverse Link Reservations that satisfy at least one of the following conditions shall not exceed one plus the value of the MaxNumReservationsRev field of MaxReservations attribute:

- The Reservation is in Open state.
- The Reservation is bound to a Stream.
- The ReservationKKQoSRequestRev corresponding to the Reservation is set to a non-default value.

7.2.5.1 Simple attributes

The negotiable simple attributes for this protocol is listed in Table 27. The access terminal and the access network shall use as defaults the values in Table 27 typed in ***bold italics***.

Table 27—Configurable values

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x01 <i>KK</i> <i>KK</i> is the two-digit hexadecimal ReservationLabel.	Reservation <i>KK</i> StreamFwd <i>KK</i> is the two-digit hexadecimal Forward Link ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters.	Soft/ Dynamic	<i>0xff</i>	No Stream associated with this Forward ReservationLabel. This is default value for all values of <i>KK</i> other than 0xfe.
			0x07	Stream 7 is associated with this Forward ReservationLabel.
			<i>0x08</i>	Stream 8 is associated with this Forward ReservationLabel. This is default value for <i>KK</i> = 0xfe.
			0x09 to 0x1e	Stream associated with this Forward ReservationLabel.
			All other values	Reserved
0x02 <i>KK</i> <i>KK</i> is the two-digit hexadecimal ReservationLabel.	Reservation <i>KK</i> StreamRev <i>KK</i> is the two-digit hexadecimal Reverse Link ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters.	Soft/ Dynamic	<i>0xff</i>	No Stream associated with this Reverse ReservationLabel. This is default value for all values of <i>KK</i> other than 0xfe.
			0x07	Stream 7 is associated with this Reverse ReservationLabel.
			<i>0x08</i>	Stream 8 is associated with this Reverse ReservationLabel. This is default value for <i>KK</i> = 0xfe.
			0x09 to 0x1e	Stream associated with this Reverse ReservationLabel.
			All other values	Reserved

Table 27—Configurable values (continued)

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x03KK KK is the two-digit hexadecimal ReservationLabel.	ReservationKKRoRAAllowedFwd KK is the two-digit hexadecimal Forward Link ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters.	Soft/Dynamic	0x00	Access terminal is not allowed to send ReservationOnRequest message for this Forward ReservationLabel.
			0x01	Access terminal is allowed to send ReservationOnRequest message for this Forward ReservationLabel.
			All other values	Reserved
0x04KK KK is the two-digit hexadecimal ReservationLabel.	ReservationKKRoRAAllowedRev KK is the two-digit hexadecimal Reverse Link ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters.	Soft/Dynamic	0x00	Access terminal is not allowed to send ReservationOnRequest message for this Reverse ReservationLabel.
			0x01	Access terminal is allowed to send ReservationOnRequest message for this Reverse ReservationLabel.
			All other values	Reserved
0x0000	NetworkInitiatedQoSAllowed	Soft/Static	0x00	Network initiated QoS is not allowed.
			0x01	Network initiated QoS is allowed.
			All other values	Reserved

7.2.5.2 Complex attributes

The following complex attributes and default values are defined (see 15.2.4 for attribute record definition).

7.2.5.2.1 ANSupportedQoSProfiles attribute

The sender shall set AttributeID field to 0x8000 (see Table 28).

Table 28—ANSupportedQoSProfiles attribute

Field	Length (bits)	Default
QoSProfileCount	8	0

QoSProfileCount of the following record:

Table 28—ANSupportedQoSProfiles attribute (continued)

Field	Length (bits)	Default
ProfileType	8	N/A
ProfileLength	8	N/A
ProfileValue	ProfileLength × 8	N/A

QoSProfileCount The sender shall set this field to the number of QoS Profiles that are included in this message.

ProfileType The sender shall set this field to 0x04. See Table 12 for encoding of ProfileType field.

ProfileLength The sender shall set this field to length of the ProfileValue field in units of octets.

ProfileValue The sender shall set this field to the profile according to 7.2.5.2.1.1.

7.2.5.2.1.1 ProfileValue definition

7.2.5.2.1.1.1 Definition of ProfileValue field when ProfileType is 0x00

The ProfileValue field shall be omitted when ProfileType is 0x00.

7.2.5.2.1.1.2 Definition of ProfileValue field when ProfileType is 0x01 or 0x02

Table 29—ProfileValue encoding when ProfileType is 0x01 or 0x02

Field	Length (bits)
FLOW_PRIORITY	4
NUM_QoS_ATTRIBUTE_SETS	3
NUM_QoS_ATTRIBUTE_SETS occurrences of the following two fields:	
QoS_ATTRIBUTE_SET_LEN	4
QoS_ATTRIBUTE_SET	8 × QoS_ATTRIBUTE_SET_LEN
Padding	0–7 bits as needed

FLOW_PRIORITY The Flow Priority Indicator. The sender shall include this field to indicate the priority to be assigned to the IP flow. The value '1111' indicates the highest priority and value '0000' indicates the lowest priority.

NUM_QoS_ATTRIBUTE_SETS

The number of QoS_ATTRIBUTE_SET. The sender shall set this field to one less than the number of alternative QoS attribute sets acceptable for the IP flow that are included. Each QoS_ATTRIBUTE_SET contains one set of acceptable QoS parameters. The sender shall include one occurrence of the following two fields for each QoS attribute set. If multiple QoS attribute sets are included, the sender shall include the QoS attribute sets in descending order of preference.

QoS_ATTRIBUTE_SET_LEN

The length of QoS_ATTRIBUTE_SET. The sender shall set this field to the length, in octets, of the QoS_ATTRIBUTE_SET.

QoS_ATTRIBUTE_SET

The QoS parameters requested for the IP flow. The sender shall include the following fields to identify the requested QoS parameters for the IP flow. The format of this field shall be as specified in 7.2.5.2.1.1.

Padding

Padding bits. The sender shall add Padding bits to make the length of the ProfileValue field an integer number of octets. The sender shall set these bits to zero. The receiver shall ignore this field.

7.2.5.2.1.1.2.1 Definition of QoS_ATTRIBUTE_SET field**Table 30—QoS_ATTRIBUTE_SET encoding**

Field	Length (bits)
QoS_ATTRIBUTE_SET_ID	7
VERBOSE	1
FlowProfileID	16 ^a
Traffic_Class	3 ^a
Peak_Rate	16 ^a
Bucket_Size	16 ^a
Token_Rate	16 ^a
Max_Latency	8 ^a
Max_IP_Packet_Loss_Rate	5 ^a
Packet_Size	8 ^a
Delay_Var_Sensitive	1 ^a
Paging_QoS_Type	2 ^a
Padding	0–7 as needed

^aThis field may be absent (see field description for conditions).

QoS_ATTRIBUTE_SET_ID

The flow QoS identifier. The sender shall set this field to an identifier for the QoS_ATTRIBUTE_SET. The sender shall not set this field to '00000000'.

VERBOSE

The VERBOSE Bit. This field indicates whether a FlowProfileID or detailed QoS parameters are included in the QoS_ATTRIBUTE_SET. If a FlowProfileID is included, the sender shall set this field to '0'. If detailed QoS parameters are included, the sender shall set this field to '1'.

FlowProfileID

The IP Flow's Profile Identifier. If the VERBOSE field is set to '0', the sender shall include this field and set it to the FlowProfileID that represents the application using the IP flow; otherwise, the sender shall omit this field. The FlowProfileIDs are specified in 3GPP2 C.R1001-F v1.0.

Traffic_Class

The Traffic Class. If the VERBOSE field is set to '1', the sender shall include this field to indicate the traffic class as specified in Table 31; otherwise, the sender shall omit this field.

Table 31—Traffic Class encoding

Value	Description
000	Unknown
001	Conversational
010	Streaming
011	Interactive
100	Background
101–111	Reserved

Peak_Rate

The Peak Rate. This field specifies the peak traffic rate as specified in IETF RFC 2210. If the VERBOSE field is set to '1', the sender shall include this field to indicate the peak rate, in units of 256 bytes per second; otherwise, the sender shall omit this field. If this field is included, the sender shall set this field to the unsigned value n (range from 1 to 65 535), where the peak rate = $n \times 256$ bytes per second. If the sender does not want to specify the peak rate, the sender shall set this field to zero.

Bucket_Size

The Token Bucket Size. This field specifies the token bucket size as specified in IETF RFC 2210. If the VERBOSE field is set to '1', the sender shall include this field to indicate the token bucket size, in units of 256 bytes; otherwise, the sender shall omit this field. If this field is included, the sender shall set this field to the unsigned value n (range from 1 to 65 535), where the bucket size = $n \times 256$ bytes. If the sender does not want to specify the token bucket size, the sender shall set this field to zero.

Token_Rate

The Token Rate. This field specifies the token rate as specified in IETF RFC 2210. If the VERBOSE field is set to '1', the sender shall include this field to indicate the token rate, in units of 256 bytes per second; otherwise, the sender shall omit this field. If this field is included, the sender shall set this field to the unsigned value n (range from 1 to 65 535), where the token rate = $n \times 256$ bytes per second. If the sender does not want to specify the token rate, the sender shall set this field to zero.

Max_Latency

The Maximum Latency. If the VERBOSE field is set to '1', the sender shall include this field to indicate the maximum latency, in units of 10 ms; otherwise, the sender shall omit this field. Max Latency specifies the maximum acceptable delay from the time that an octet of user data is submitted to the transmitter until the receiver receives the octet. It is measured between the access terminal and access network. If this field is included, the sender shall set this field to the unsigned value n (range from 1 to 255), where the maximum latency = $n \times 10$ ms. If the sender does not want to specify the maximum latency, the sender shall set this field to zero.

Max_IP_Packet_Loss_Rate

The Maximum Loss Rate. If the VERBOSE field is set to '1', the sender shall include this field to indicate the maximum IP packet loss rate; otherwise, the

sender shall omit this field. If this field is included, the sender shall set this field to an unsigned value n (range from 1 to 31), where the maximum IP packet loss rate = $10^{(-n/4)}$. If the sender does not want to specify the maximum loss rate, the sender shall set this field to zero. When Max_IP-Packet_loss_Rate is used the Packet_Size shall not be set to zero.

Packet_Size The Median Packet Size. If the VERBOSE field is set to '1', the sender shall include this field to indicate the median packet size, in units of 8 bytes; otherwise, the sender shall omit this field. If this field is included, the sender shall set this field to the unsigned value n (range from 1 to 255), where the median packet size = $n \times 8$ bytes. If the sender does not want to specify the median packet size, the sender shall set this field to zero.

Delay_Var_Sensitive The delay variation sensitivity indicator. If the VERBOSE field is set to '1', the sender shall include this field; otherwise, the sender shall omit this field. If this field is included, the sender shall set this field to '1' to indicate that the traffic flow is sensitive to variation in delay. The sender shall set this field to '0' to indicate the traffic flow sensitivity to variation in delay is not specified.

Paging_QoS_Type The Paging QoS Type. If the VERBOSE field is set to '1', the sender shall include this field; otherwise, the sender shall omit this field.

If this field is included, the sender shall set this field as follows: Set to '00' to indicate that the traffic flow is sensitive to paging delay, set to '01' to indicate that the traffic flow is tolerant to paging delay but buffer the data, set to '10' to indicated not to page for the traffic flow and discard the data.

Padding Padding bits. The sender shall add Padding bits to make the length of the QoS_ATTRIBUTE_SET field an integer number of octets. The sender shall set these bits to zero. The receiver shall ignore this field.

7.2.5.2.1.1.3 Definition of ProfileValue field when ProfileType is 0x03

Table 32—ProfileValue when ProfileType is 0x03

Field	Length (bits)
QoS_ATTRIBUTE_SET_ID	7
Reserved	1

QoS_ATTRIBUTE_SET_ID The identifier assigned to the QoS_ATTRIBUTE_SET that has been granted by the access network. The access network shall set this field to the identifier assigned to the QoS_ATTRIBUTE_SET that have been granted by the access network.

Reserved The Reserved bit. The receiver shall ignore this field.

7.2.5.2.1.1.4 Definition of ProfileValue field when ProfileType is 0x04

The ProfileValue field shall be set to the FlowProfileID as specified in 3GPP2 C.R1001-F v1.0.

Commit	Soft	Scope	Dynamic

7.2.5.2.2 ATSupportedQoSProfiles attribute

The sender shall set AttributeID field to 0x8001 (see Table 33).

Table 33—ATSupportedQoSProfiles attribute

Field	Length (bits)	Default
QoSProfileCount	8	0
QoSProfileCount of the following record:		
ProfileType	8	N/A
ProfileLength	8	N/A
ProfileValue	ProfileLength × 8	N/A

QoSProfileCount The sender shall set this field to the number of QoS Profiles that are included in this message.

ProfileType The sender shall set this field to 0x04. See Table 12 for encoding of ProfileType field.

ProfileLength The sender shall set this field to length of the ProfileValue field in units of octets.

ProfileValue The sender shall set this field to the profile according to 7.2.5.2.1.1.1.

Commit	Soft

Scope	Static

7.2.5.2.3 ReservationKKQoSRequestFwd attribute

KK is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters (see Table 34).

The sender shall set AttributeID field to 0x81*KK*, where *KK* is the two-digit hexadecimal ReservationLabel.

Table 34—ReservationKKQoSRequestFwd attribute

Field	Length (bits)	Default
ProfileType	8	<i>0x00</i>
ProfileLength	16	<i>0x0000</i>
ProfileValue	ProfileLength × 8	N/A

ProfileType The sender shall set this field to 0x01 or 0x02. See Table 12 for encoding of ProfileType field.

ProfileLength The sender shall set this field to length of the ProfileValue field in units of octets.

ProfileValue The sender shall set this field to the Profile Value corresponding to the Profile Type according to 7.2.5.2.1.1.1.

Commit	Soft

Scope	Static

7.2.5.2.4 ReservationKKQoSRequestRev attribute

KK is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters (see Table 35).

The sender shall set AttributeID field to 0x82KK, where KK is the two-digit hexadecimal ReservationLabel.

Table 35—ReservationKKQoSRequestRev attribute

Field	Length (bits)	Default
ProfileType	8	0x00
ProfileLength	16	0x0000
ProfileValue	ProfileLength × 8	N/A

ProfileType The sender shall set this field to 0x01 or 0x02. See Table 12 for encoding of ProfileType field.

ProfileLength The sender shall set this field to length of the ProfileValue field in units of octets.

ProfileValue The sender shall set this field to the Profile Value corresponding to the Profile Type according to 7.2.5.2.1.1.1.

Commit	Soft

Scope	Static

7.2.5.2.5 MaxReservations attribute

The sender shall set AttributeID field to 0x8002 (see Table 36).

Table 36—MaxReservations attribute

Field	Length (bits)	Default
MaxNumReservationsFwd	8	0x0f
MaxNumReservationsRev	8	0x0f
MaxNumOpenReservationsFwd	8	0x0f
MaxNumOpenReservationsRev	8	0x0f

MaxNumReservationsFwd

The sender shall set this field to indicate one less than the maximum total number of Forward Link Reservations supported that satisfy at least one of the following conditions:

- The Reservation is in Open state.
- The Reservation is bound to a Stream.
- The ReservationKKQoSRequestFwd corresponding to the Reservation is set to a non-default value.

The value shall be in the range of 0x00 to 0xfe, inclusive.

MaxNumReservationsRev

The sender shall set this field to indicate one less than the maximum total number of Reverse Link Reservations supported that satisfy at least one of the following conditions:

- The Reservation is in Open state.
- The Reservation is bound to a Stream.
- The ReservationKKQoSRequestRev corresponding to the Reservation is set to a non-default value.

The value shall be in the range of 0x00 to 0xfe, inclusive.

MaxNumOpenReservations

Fwd The sender shall set this field to indicate one less than the maximum number of simultaneous Open Forward Link Reservations supported. The value shall be in the range of 0x00 to MaxNumReservationsFwd, inclusive.

MaxNumOpenReservationsRev

The sender shall set this field to indicate one less than the maximum number of simultaneous Open Reverse Link Reservations supported. The value shall be in the range of 0x00 to MaxNumReservationsRev, inclusive.

Commit	Soft

Scope	Static

7.2.5.2.6 ReservationKKPacketFiltersFwd attribute

KK is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters (see Table 37).

The sender shall set AttributeID field to 0x83KK, where KK is the two-digit hexadecimal ReservationLabel with valid range of 0x00 to 0xfd.

Table 37—ReservationKKPacketFiltersFwd attribute

Field	Length (bits)	Default
NumPacketFilters	8	0x00
NumPacketFilters occurrences of the following record:	—	—
EvaluationPrecedence	8	N/A
PacketFilterLength	16	N/A
PacketFilter	PacketFilterLength × 8	N/A

NumPacketFilters The sender shall set this field to the number of packet filters included in this message.

EvaluationPrecedence The higher the value of the packet filter evaluation precedence field, the lower the precedence of that packet filter. Value of 0xff is used as an indication of no precedence. The sender shall set this field to the packet filter evaluation precedence. The sender shall set this field either to a value that is not used by EvaluationPrecedence field of any of the ReservationKKPacketFiltersFwd attributes or to 0xff.

PacketFilterLength The sender shall set this field to the length of the PacketFilter field in octets.

PacketFilter The sender shall set this field to the packet filter. Packet filter format is specified in 7.2.5.2.6.1.

Commit	Soft	Scope	Static

7.2.5.2.6.1 Definition of PacketFilter record

Table 38—PacketFilter record

Field	Length (bits)	Default
PacketFilterType	8	N/A

One or more occurrences of the following record:

PacketFilterComponentTypeID	8	N/A
PacketFilterComponent	Variable	N/A

PacketFilterType The sender shall set this field to 0x00 to indicate packet filter applies to outer IP header. The sender shall set this field to 0x01 to indicate packet filter applies to innermost encapsulated IP header.

PacketFilterComponentTypeID

The sender shall set this field to identifier of this packet filter component as specified in Table 39. A given packet filter component type identifier shall not appear more than once inside packet filter content. Among the “IPv4 Source Address” and “IPv6 Source Address” packet filter components, only one shall be present in one packet filter. Among the “single destination port” and “destination port range” packet filter components, only one shall be present in one packet filter. Among the “single source port” and “source port range” packet filter components, only one shall be present in one packet filter.

Table 39—PacketFilterComponentTypeID encoding

PacketFilterComponentTypeID	Length (octets) of PacketFilterComponent	Meaning
0x10	5	IPv4 Source Address with Prefix Length
0x20	17	IPv6 Source Address with Prefix Length
0x11	5	IPv4 Destination Address with Prefix Length
0x21	17	IPv6 Destination Address with Prefix Length
0x30	1	Protocol /Next header
0x40	2	Single Destination Port
0x41	4	Destination Port range
0x50	2	Single Source Port
0x51	4	Source Port range
0x60	4	Security Parameter Index

Table 39—PacketFilterComponentTypeID encoding (continued)

PacketFilterComponentTypeID	Length (octets) of PacketFilterComponent	Meaning
0x70	2	Type of Service/Traffic Class
0x80	3	Flow label
0x81	17	Type 2 Routing Header with Prefix Length
0x82	17	Home Address Option with Prefix Length

PacketFilterComponent The IPv4 Source Address shall be encoded as a sequence of a four octet IPv4 Source Address followed by a one octet Prefix Length field. The address up to the Prefix Length is matched against the IPv4 Source Address in the IPv4 header of the packet. If packet filter is setup for the reverse direction, the prefix length shall be set to 32.

The IPv6 Source Address,¹⁴ packet filter component value field shall be encoded as a sequence of a sixteen octet IPv6 Source Address field followed by a one octet Prefix Length field. The address up to the Prefix Length is matched against the IPv6 Source Address in the IPv6 header of the packet. If packet filter is setup for the reverse direction, the prefix length shall be set to 64. The AT should not include a source address packet filter component unless it is reasonably sure that the IP address of the correspondent node will not change for the life of the application flow or is willing to update the packet filters when such a change takes place.

The IPv4 Destination Address packet filter component shall be encoded as a sequence of a four octets IPv4 Destination Address field followed by a one octet Prefix Length field. The address up to the Prefix Length is matched against the IPv4 Destination Address in the IPv4 header of the packet. If packet filter is setup for the forward direction, the prefix length shall be set to 32.

The IPv6 Destination Address packet filter component shall be encoded as a sequence of a sixteen-octet IPv6 address field followed by a one octet Prefix Length field. The address up to the Prefix Length is matched against the IPv6 destination Address in the IPv6 header of the packet. If packet filter is setup for the forward direction, the prefix length shall be set to 64.

The Protocol/Next Header packet filter component shall be encoded as one octet that specifies the Protocol field of the IPv4 header or IPv6 Next Header. The value range is from 0 to 255.

The Single Destination Port and Single Source Port packet filter component value field shall be encoded as two octets that specify a port number. Port numbers range between 0 and 65 535.

The Destination Port range and Source Port range packet filter component value field shall be encoded as a sequence of two octets for the lower limit of the range followed by two octets for the higher limit of the range. Port numbers range between 0 and 65 535.

¹⁴Source addresses of forward direction packet flows belong to correspondent nodes and are sometimes subject to change due to, e.g., renumbering [IETF RFC 2461] or privacy [IETF RFC 3041].

The Security Parameter Index packet filter component field shall be encoded as four octets that specify the IPsec SPI.

The Type of Service (IPv4)/Traffic Class (IPv6) packet filter component shall be encoded as a sequence of a one octet Type of Service/Traffic Class field and a one octet Type of Service/Traffic Class mask field. The Type of Service /Traffic Class mask determines which bits of the Type of Service or Traffic Class octet shall be used to match against the actual value of the corresponding field in the IP packets. The mask contains ones in the bit positions to be used in the matching operation.

The Flow label packet filter component shall be encoded as three octets that specify the IPv6 flow label. The bits 0 through 3 of the first octet shall be used for padding whereas the remaining 20 bits shall contain the IPv6 flow label as per IETF RFC 2460.

The Type 2 Routing Header packet filter component shall be encoded as a sequence of a sixteen-octet IPv6 address field followed by a one octet Prefix Length field. The address up to the Prefix Length is matched against the Type 2 Routing Header in the outer IPv6 header of the packet. If the value of packet filterType is 0, this packet filter component may be included.

The Home address Option packet filter component shall be encoded as a sequence of a sixteen-octet IPv6 address field followed by a one octet Prefix Length field. The address up to the Prefix Length is matched against the Home address in the outer IPv6 header of the packet. If the value of packet filter Type is 0, this packet filter component may be included.

7.2.5.2.7 ReservationKKPacketFiltersRev attribute

KK is the two-digit hexadecimal ReservationLabel, where hexadecimal digits A through F are specified in uppercase letters.

The sender shall set AttributeID field to 0x84KK, where *KK* is the two-digit hexadecimal ReservationLabel with valid range of 0x00 to 0xfd (see Table 40).

Table 40—ReservationKKPacketFiltersRev attribute

Field	Length (bits)	Default
NumPacketFilters	8	0x00
NumPacketFilters occurrences of the following record:		
EvaluationPrecedence	8	N/A
PacketFilterLength	16	N/A
PacketFilter	PacketFilterLength × 8	N/A

NumPacketFilters The sender shall set this field to the number of packet filters included in this message.

EvaluationPrecedence The higher the value of the packet filter evaluation precedence field, the lower the precedence of that packet filter. Value of 0xff is used as an indication of no precedence. The sender shall set this field to the packet filter evaluation

precedence. The sender shall set this field either to a value that is not used by EvaluationPrecedence field of any of the ReservationKKPacketFiltersRev attributes or to 0xff.

- PacketFilterLength The sender shall set this field to the length of the PacketFilter field in octets.
- PacketFilter The sender shall set this field to the packet filter. Packet filter format is specified in 7.2.5.2.6.1.

Commit	Soft	Scope	Static

7.2.6 Non-attribute data

This protocol does not define any non-attribute data.

7.2.7 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

All configuration attributes and non-attribute data are Session State Information records. This protocol does not define additional parameter records.

7.3 Basic Radio Link Protocol

7.3.1 Introduction

7.3.1.1 General overview

The Basic Radio Link Protocol provides a packet stream that can be used to carry packets between the access terminal and the access network. The packet stream is called a Stream.

The relationship between the Basic Radio Link Protocol sub-protocols is illustrated in Figure 23.

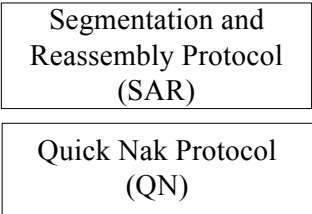


Figure 23—Basic Radio Link Protocol Sub-Protocols

Figure 24 illustrates the relationship for Stream between the Basic Radio Link Protocol and other protocols. The protocols defined in the Basic Radio Link Protocol are shown shaded.

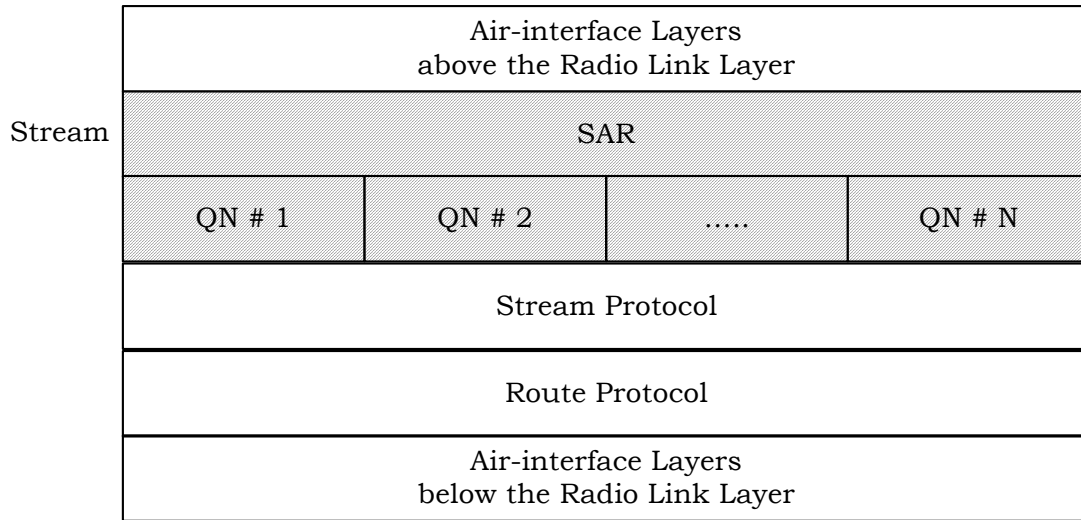


Figure 24—Relationship between Basic Radio Link Protocol and other protocols

The Basic Radio Link Protocol provides the following:

- The Segmentation and Reassembly (SAR) Sub-Protocol, which provides segmentation and reassembly, retransmission (if needed), and duplicate detection of higher-layer packets (if needed).
- The Quick Nak Sub-Protocol, which provides indication of erased higher-layer packets.
- The protocol subtype for this protocol is defined in 16.1. This protocol uses the Signaling Protocol to transmit and receive messages.

7.3.1.2 Local common data

This protocol does not define any local common data.

7.3.1.3 Public data

- Subtype for this protocol
- Stream Number
- ActivatedFwd
- ActivatedRev
- TransmitAbortTimerRev
- All data defined as Static Attribute, Static Non-Attribute Data, and local common data

7.3.1.4 Data encapsulation for the InUse instance of the protocol

Figure 25 illustrates the relationship between packets from the Higher Layer Protocol, RLP packets, and Stream Protocol payload. The transmission unit of this protocol is an RLP packet.

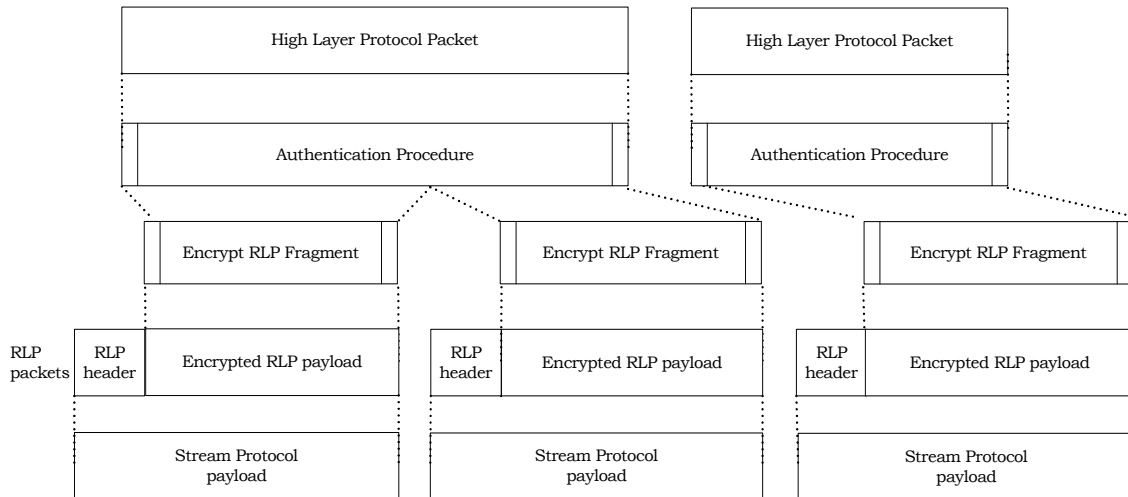


Figure 25—Basic Radio Link Protocol encapsulation

7.3.2 Procedures and messages for the InConfiguration instance of the protocol

7.3.2.1 Protocol initialization for the InConfiguration Protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

7.3.2.2 Procedures

If the Stream number corresponding to this protocol is between 0x00 through 0x06 (both inclusive) then the access network and the access terminal shall not negotiate ProtocolParametersFwd and ProtocolParametersRev attributes.

If Stream number corresponding to this protocol is 0x00, 0x01, 0x02, 0x03, or 0x04, then the access network and the access terminal shall not negotiate ActivatedFwd attribute. If Stream number corresponding to this protocol is 0x01, 0x02, or 0x04, then the access network and the access terminal shall not negotiate ActivatedRev attribute.

Access network and access terminal shall not negotiate following combination of attribute values:

- AutoSyncReceiverFwd equal to 0x01 and NakAckEnableFwd equal to value other than 0x00.
- AutoSyncReceiverRev equal to 0x01 and NakAckEnableRev equal to value other than 0x00.

7.3.2.3 Message formats

This protocol does not define any messages.

7.3.3 Common procedures and messages for the InUse instance of the protocol

7.3.3.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and the access network shall perform the following in the order specified:

- Perform the procedures specified in 15.1.2.
- If Stream number corresponding to this protocol is between 0x00 through 0x06 (both inclusive) then the access network and the access terminal shall set ProtocolParametersFwd attributes as shown in Table 41.

Table 41—ProtocolParametersFwd attribute values for Streams 0x00 to 0x06

Stream	ProtocolParametersFwd		
	ProtocolID	ProtocolTypeInUse	ProtocolTypeInConfig
0x00	0x06	0xa006	0xe006
0x01	0x06	0xa006	0xe006
0x02	0x06	0xa006	0xe006
0x03	0x07	0xa007	0xe007
0x04	0x07	0xa007	0xe007
0x05	0x07	0xa007	0xe007
0x06	0x07	0xa007	0xe007

- If Stream number corresponding to this protocol is between 0x00 through 0x06 (both inclusive) then the access network and the access terminal shall set ProtocolParametersRev attributes as shown in Table 42.

Table 42—ProtocolParametersRev attribute values for Streams 0x00 to 0x06

Stream	ProtocolParametersRev		
	ProtocolID	ProtocolTypeInUse	ProtocolTypeInConfig
0x00	N/A	N/A	N/A
0x01	0x06	0xa006	0xe006
0x02	0x06	0xa006	0xe006
0x03	N/A	N/A	N/A
0x04	0x07	0xa007	0xe007
0x05	0x07	0xa007	0xe007
0x06	0x07	0xa007	0xe007

- Reset SARResetCounter to zero.
- Perform the initialization procedures defined in 7.3.4.3.1.1.1.1 and 7.3.4.3.1.1.1.2 for all activated Streams.

7.3.3.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

7.3.3.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

7.3.3.4 Procedures

7.3.3.4.1 RouteCounter procedures

The access network and the access terminal shall maintain a 15-bit variable RouteCounter.

Upon receiving RouteControl.RouteCreated indication, the access network and access terminal shall initialize RouteCounter to the value of RouteCounter provided in the indication.

7.3.3.4.2 RLP Transmit procedures

The RLP receives packets for transmission from the Higher Layer Protocol, if necessary the RLP adds an authentication tag by calling AUTHENTICATE_ADD_TAG procedure of the Message Integrity Protocol, if necessary it fragments the received packet, if necessary it encrypts the fragment by calling the ENCRYPT procedure of the Ciphering Protocol and forms an RLP packet by prepending the RLP packet header defined in 7.3.3.4.5 to the fragment.

RLP shall form the Authentication sub-packet from the higher-layer packet as specified in 7.3.3.4.4.1.

If one of the following conditions is true:

- This Stream is 0x00 (forward link), 0x03 (forward link), 0x04, 0x05, or 0x06.
- This Stream is 0x01, or 0x02, and the Higher Layer Protocol indicates that payload is not to be encrypted.

then RLP shall form RLP packet specified as follows:

- SAR sub-protocol of RLP shall divide Authentication sub-packet in to one or more fragments.
- RLP shall prepend RLP packet header to each of the fragments to form an RLP packet. For each of the RLP packet carrying the Authentication sub-packet, RLP shall set CipheringKeyIndex field of the RLP packet header to '00'.

Otherwise, RLP shall form RLP packet specified as follows:

- SAR sub-protocol of RLP shall divide Authentication sub-packet in to one or more fragments.
- RLP shall call ENCRYPT procedure of Ciphering Protocol for each fragment with its inputs set as follows:
 - If transmitter is access network, then set the Direction to ForwardLink. Otherwise, set the *Direction* to ReverseLink.
 - If transmitter is access network, then set the DataUnit to DataUnitFwd. Otherwise, set the *DataUnit* to DataUnitRev.
 - Set the *StreamID* to ('0000 0000 000' | Stream Number).
 - Set the *RouteCounter* to RouteCounter.
 - Set the *SARResetCounter* to SARResetCounter.

- Set the *SARSequenceNumber* to SAR sequence number of first data unit of the fragment.
- Set the *SARSequenceRolloverCounter* to *SARSequenceRolloverCounter* associated with the SAR sequence number of the first data unit of the fragment.
- Set the *Payload* to the address of the beginning of the memory space that contains the fragment.
- Set the *PayloadSize* to the length of the fragment in octets.
- For each of the fragments returned from the ENCRYPT procedure, the RLP shall form RLP packet as follows:
 - The RLP shall set the RLP packet payload to the output of the ENCRYPT procedure, which starts at the memory space specified by *Payload* and is of the same size as the original fragment.
 - RLP shall prepend RLP packet header to the RLP packet payload.
 - RLP shall set *CipheringKeyIndex* field of the RLP packet header to *CipheringKeyIndex* output of the ENCRYPT procedure.

7.3.3.4.3 RLP Receive procedures

SAR sub-protocol maintains a resequencing buffer. If necessary, RLP decrypts the received data units before adding them to resequencing buffer. If *CipheringKeyIndex* field of the RLP packet header is set to '00', then RLP shall store the received data units in the resequencing buffer without performing decryption operation. Otherwise, RLP shall decrypt the received data units specified as follows and then store decrypted data units in the resequencing buffer:

- RLP shall call DECRYPT procedure of Ciphering Protocol with its inputs set as follows:
 - If receiver is access network, then set the *Direction* to *ReverseLink*. Otherwise, set the *Direction* to *ForwardLink*.
 - If receiver is access network, then set the *DataUnit* to *DataUnitRev*. Otherwise, set the *DataUnit* to *DataUnitFwd*.
 - Set the *StreamID* to ('0000 0000 000' | Stream Number).
 - Set the *RouteCounter* to *RouteCounter*.
 - Set the *SARResetCounter* to *SARResetCounter*.
 - Set the *SARSequenceNumber* to SAR sequence number of first data unit of the RLP payload.
 - Set the *SARSequenceRolloverCounter* to *SARSequenceRolloverCounter* associated with the SAR sequence number of first data unit of the RLP payload.
 - Set the *Payload* to the address of the beginning of the memory space that contains the RLP payload.
 - Set the *PayloadSize* to the length of the RLP payload in octets.
 - Set *CipheringKeyIndex* to *CipheringKeyIndex* field of RLP header.
- After the DECRYPT procedure is returned, the decrypted data units are available at the output of the DECRYPT procedure, which starts at the memory space specified by *Payload* and is of the same size as the original RLP payload.

Higher Layer Protocol packets described in 7.3.4.3.2.2 are passed to the Higher Layer Protocol only after performing authentication procedures specified in 7.3.3.4.4.2. RLP shall form Authenticated Higher Layer Protocol packet from Higher Layer Protocol packet as specified in 7.3.3.4.4.2, and shall pass Authenticated Higher Layer Protocol packet to the Higher Layer Protocol.

7.3.3.4.4 Authentication procedures

7.3.3.4.4.1 Transmit procedures

If this Stream is 0x01 or 0x02, then RLP shall form the Authentication sub-packet from the higher-layer packet specified as follows:

- RLP shall call AUTHENTICATE_ADD_TAG procedure of the Message Integrity Protocol with its inputs set as follows:
 - If transmitter is access network, then set the *Direction* to ForwardLink. Otherwise, set the *Direction* to ReverseLink.
 - Set the *StreamID* to ('0000 0000 000' | Stream Number).
 - Set the *RouteCounter* to RouteCounter.
 - Set the *SARResetCounter* to SARResetCounter.
 - Set the *SARSequenceNumber* to SAR sequence number of first data unit of this Authentication sub-packet.
 - Set the *SARSequenceRolloverCounter* to SARSequenceRolloverCounter associated with the SAR sequence number of the first data unit of this Authentication sub-packet.
 - Set the *InputPayload* to the address of the beginning of the memory space that contains the higher-layer packet.
 - Set the *InputPayloadSize* to the length of the higher-layer packet in octets.
- After the AUTHENTICATE_ADD_TAG procedure is returned, the RLP shall set the Authentication sub-packet to the output of the AUTHENTICATE_ADD_TAG procedure, which starts at the memory space specified by OutputPayload and is of the size specified by OutputPayloadSize.

Otherwise, RLP shall form Authentication sub-packet specified as follows:

- The RLP shall set the Authentication sub-packet to the higher-layer packet.

7.3.3.4.4.2 Receive procedures

If this Stream is 0x01 or 0x02, then RLP shall form Authenticated Higher Layer Protocol packet from Higher Layer protocol packet specified as follows:

- RLP shall call AUTHENTICATE_CHECK_TAG procedure of Message Integrity Protocol with its inputs set as follows:
 - If receiver is access network, then set the *Direction* to ReverseLink. Otherwise, set the *Direction* to ForwardLink.
 - Set the *StreamID* to ('0000 0000 000' | Stream Number).
 - Set the *RouteCounter* to RouteCounter.
 - Set the *SARResetCounter* to SARResetCounter.
 - Set the *SARSequenceNumber* to SAR sequence number of first data unit of this Higher Layer Protocol packet.
 - Set the *SARSequenceRolloverCounter* to SARSequenceRolloverCounter associated with the SAR sequence number of the first data unit of this Higher Layer Protocol packet.
 - Set the *InputPayload* to the address of the beginning of the memory space that contains the Higher Layer Protocol packet.
 - Set the *InputPayloadSize* to the length of the Higher Layer Protocol packet in octets.
- After the AUTHENTICATE_CHECK_TAG procedure is returned, if the TagMatched output is set to FALSE, then the RLP shall discard the Higher Layer Protocol packet. Otherwise, the RLP shall set Authenticated Higher Layer Protocol packet to the output of the

AUTHENTICATE_CHECK_TAG procedure, which starts at the memory space specified by OutputPayload and is of the size specified by OutputPayloadSize.

Otherwise, RLP shall form Authenticated Higher Layer Protocol packet specified as follows:

- The RLP shall set the Authenticated Higher Layer Protocol packet to the Higher Layer Protocol packet.

7.3.3.4.5 RLP packet header

The RLP packet header, which precedes the RLP payload, has the following format (see Table 43):

Table 43—RLP packet header

Field	Length (bits)
CipheringKeyIndex	2
SARSEQ	Indicated by SequenceLengthFwd or SequenceLengthRev
FirstDataUnit	1
LastDataUnit	1
QNSEQValid	1 ^a
QNSEQ	7 ^a

^aThis field may be absent (see field description for conditions).

CipheringKeyIndex Sender shall set this field to indicate index of the key used for encryption of the payload of this RLP packet as specified in Table 44. Ciphering Protocol provides value of this field.

Table 44—CipheringKeyIndex encoding

CipheringKeyIndex	Meaning
'00'	RLP packet is not encrypted
'01'	RLP packet encrypted with key corresponding to index '01'
'10'	RLP packet encrypted with key corresponding to index '10'
'11'	RLP packet encrypted with key corresponding to index '11'

SARSEQ The SAR sequence number of the first data unit in the RLP payload. If this RLP packet is being sent on the forward link, the access network shall set the length of this field according to the SequenceLengthFwd attribute corresponding to this Stream. If this RLP packet is being sent on the reverse link, the access terminal shall set the length of this field according to the SequenceLengthRev attribute corresponding to this Stream. The SARSEQ is either 6, 14, 22, or 30 bits in length.

FirstDataUnit	If the payload of this RLP packet is the first segment of a Higher Layer Protocol packet, then the sender shall set this field to '1'. Otherwise, the sender shall set this field to '0'.
LastDataUnit	If the payload of this RLP packet is the last segment of a Higher Layer Protocol packet, then the sender shall set this field to '1'. Otherwise, the sender shall set this field to '0'.
QNSEQValid	The access terminal shall omit this field. If the QNEnableFwd attribute is set to 0x00, then the access network shall omit this field. Otherwise, the access network shall include this field and set it as follows: If the QNSEQ included in this header is valid, then the sender shall set this field to '1'. Otherwise, the sender shall set this field to '0'. If this RLP packet is being sent on the forward link, and if the payload of this RLP packet is being retransmitted, then the access network shall set this field to '0'.
QNSEQ	The access terminal shall omit this field. If the QNEnableFwd attribute is set to 0x00, then the access network shall omit this field. Otherwise, the access network shall include this field and set it as follows: If the QNSEQValid field is set to '1', then the access network shall set this field to $V(S)_{QN}$. Otherwise, the access network shall set this field to '000 0000'.

7.3.4 Segmentation and Reassembly Sub-Protocol

7.3.4.1 Overview

The Segmentation and Reassembly Sub-Protocol (SAR) provides one packet stream with an acceptably low erasure rate for efficient operation of higher layer protocols (e.g., TCP). When used as part of the Basic Radio Link Protocol, the protocol carries one packet stream from the higher layer. SAR is a sub-protocol associated with the Basic Radio Link Protocol.

7.3.4.2 Primitives and public data

7.3.4.2.1 Commands

This sub-protocol does not define any commands.

7.3.4.2.2 Return indications

This sub-protocol returns the following indications:

- *ReliablePacketDeliveryFailed*
- *ReliableStreamReset*

7.3.4.3 Procedures and messages for the InUse instance of the sub-protocol

A Forward Stream is defined to be activated if the ActivatedFwd attribute is set to 0x01.

A Reverse Stream is defined to be activated if the ActivatedRev attribute is set to 0x01.

A Stream is defined to be deactivated if it is not activated.

This subclause defines the procedures and messages for the in-use instance of forward or reverse Stream.

7.3.4.3.1 Procedures

If the OutOfOrderDeliveryFwd attribute of forward Stream is 0x00, then forward Stream delivers payload to the corresponding instance of the Higher Layer Protocol in order. If the OutOfOrderDeliveryFwd attribute of forward Stream is 0x01, then forward Stream may deliver payload to the corresponding instance of the Higher Layer Protocol out of order.

The policy SAR follows in determining the number of octets to send in an RLP packet is beyond the scope of this standard. It is subject to the following requirements:

- The size of an RLP packet shall not exceed the maximum payload length that can be carried by a Radio Link Layer packet given the target channel and current transmission rate on that channel.
- An RLP packet shall contain octets from no more than one Higher Layer Protocol packet.
- If this is not a Reverse Link Serving Route (as indicated by the RLSS public data of the Reverse Control Channel MAC Protocol) and if the Higher Layer Protocol packet is transmitted for the first time, then the RLP packet sent on reverse link shall contain all unsent octets of the Higher Layer Protocol packet.
- If this is not a Forward Link Serving Route (as indicated by the FLSS public data of the Forward Control Channel MAC Protocol) and if the Higher Layer Protocol packet is transmitted for the first time, then the RLP packet sent on forward link shall contain all unsent octets of the Higher Layer Protocol packet.
- The RLP packet should contain all octets of the Higher Layer Protocol packet if all of the following conditions are true:
 - The size of the RLP packet carrying all octets of the Higher Layer Protocol packet does not exceed the maximum payload length that can be carried by a Radio Link Layer packet given the target channel and the current transmission rate on that channel.
 - The data unit for the Stream is RLP packet payloads.
- If all of the following conditions are true, the size of the RLP packet should¹⁵ be such that it could be retransmitted in a lower layer packet with same or larger payload:
 - The data unit for the Stream is RLP packet payloads.
 - If the SAR transmitter is the access terminal, NakAckEnableRev attribute is not set to 0x00 for this Stream or PhysicalLayerNakEnableRev is set to 0x01 for this Stream.
 - If the SAR transmitter is the access network, NakAckEnableFwd attribute is not set to 0x00 for this Stream.

SAR makes use of the ResetTxIndication, ResetRxIndication, ResetTxComplete, ResetRxComplete, Flush, ATReceiverStatus, and ANReceiverStatus messages to perform control related operations.

When forward Stream is activated, the access network and the access terminal shall not update the following attributes:

- AutoSyncReceiverFwd
- NakAckEnableFwd
- ProtocolParametersFwd
- SequenceLengthFwd
- DataUnitFwd
- OutOfOrderDeliveryFwd
- QNEnableFwd

¹⁵This is a “should” requirement because if RLP packet could not be retransmitted in lower layer packet, then RLP packet can be tunneled to the same route.

The access network shall not initiate re-activation of a deactivated forward Stream if the access terminal initiated deactivation of the forward Stream since the last time the Stream was activated, and the deactivation was successful. The access terminal shall not initiate re-activation of a deactivated forward Stream if the access network initiated deactivation of the forward Stream since the last time the Stream was activated, and the deactivation was successful.

When reverse Stream is activated, the access network and the access terminal shall not update the following attributes:

- AutoSyncReceiverRev
- NakAckEnableRev
- MaxTransmissionsRev
- ProtocolParametersRev
- SequenceLengthRev
- DataUnitRev

The access network shall not initiate re-activation of a deactivated reverse Stream if the access terminal initiated deactivation of the reverse Stream since the last time the Stream was activated, and the deactivation was successful. The access terminal shall not initiate re-activation of a deactivated reverse Stream if the access network initiated deactivation of the reverse Stream since the last time the Stream was activated, and the deactivation was successful.

The ProtocolID field of the ProtocolParametersFwd attribute shall be set to a value that is supported by the access terminal as indicated in the ATSupportedApplicationProtocolIDs public data of the Session Control Protocol. The ProtocolID field of the ProtocolParametersRev attribute shall be set to a value that is supported by the access terminal as indicated in the ATSupportedApplicationProtocolIDs public data of the Session Control Protocol.

If the DataUnitFwd attribute of forward Stream is 0x00, then the data unit for the Stream shall be octets. Otherwise, the data unit for the Stream shall be RLP packet payloads. If the DataUnitRev attribute of reverse Stream is 0x00, then the data unit for the Stream shall be octets. Otherwise the data unit for the Stream shall be RLP packet payloads.

7.3.4.3.1.1 Initialization and reset

The SAR initialization procedure initializes the SAR variables and data structures in one end of the link. The SAR reset procedure guarantees that SAR state variables on both sides are synchronized. The reset procedure includes initialization.

The access network shall perform the initialization procedure defined in 7.3.4.3.1.1.1 for forward Stream when forward Stream is activated. The access terminal shall perform the initialization procedure defined in 7.3.4.3.1.1.2 for forward Stream when forward Stream is activated. The access terminal shall perform the initialization procedure defined in 7.3.4.3.1.1.1 for reverse Stream when reverse Stream is activated. The access network shall perform the initialization procedure defined in 7.3.4.3.1.1.2 for reverse Stream when reverse Stream is activated.

7.3.4.3.1.1.1 Initialization procedure

All operations and comparisons performed on SARResetCounter shall be carried out in unsigned modulo 2^P arithmetic, where P is 8. For any SAR sequence number N , the sequence numbers in the range $[N + 1, N + 2^{P-1} - 1]$ shall be considered greater than N and the sequence numbers in the range $[N - 2^{P-1}, N - 1]$ shall be considered smaller than N .

7.3.4.3.1.1.1 Initialization procedure for the SAR transmitter

When SAR transmitter performs the initialization procedure it shall

- Reset the send state variable $V(S)$ to zero
- Reset SARSequenceRolloverCounter to zero
- Increment SARResetCounter by 1 modulo 2^P
- Send a *QN.Initialize* command to all instances of the QN transmitter of Stream
- Clear the retransmission queues

7.3.4.3.1.1.2 Initialization procedure for the SAR receiver

When SAR receiver performs the initialization procedure it shall

- Reset the receive state variables $V(R)$ and $V(N)$ to zero
- Reset SARSequenceRolloverCounter to zero
- Increment SARResetCounter by 1 modulo 2^P
- Send a *QN.Initialize* command to all instances of the QN receiver of Stream
- Clear the resequencing buffer

7.3.4.3.1.1.2 Reset procedure

SAR makes use of the ResetTxIndication, ResetRxIndication, ResetTxComplete, and ResetRxComplete messages to perform reset related procedures. If this Stream is not 0x02, then the SAR transmitter and the SAR receiver shall send these messages using Reliable RLP; otherwise, the SAR transmitter and the SAR receiver shall send these messages using Best Effort RLP.

7.3.4.3.1.1.2.1 Reset procedure for the initiating side when it is a SAR transmitter

If the side initiating a reset procedure is a SAR transmitter for the Stream being reset, then enter the SAR Reset State.

Upon entering the SAR Reset state SAR transmitter shall

- Perform the SAR transmitter initialization procedure defined in 7.3.4.3.1.1.1.
- If an ATReceiverStatus or ANReceiverStatus message is received for the Stream being reset while in the SAR Reset state, the message shall be ignored.
- If the SAR transmitter that initiated the reset procedure is an access terminal, and if a *ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed* indication is received for the Stream being reset while SAR is in the Reset state, then the indication shall be ignored.
- The SAR transmitter shall not transmit SAR packets while in the SAR Reset state.
- When transmission of all the SAR packets that were being transmitted before entering the SAR Reset state is completed, the SAR transmitter shall send a ResetTxIndication message.
- If SAR receives a ResetTxComplete message for the Stream being reset while in the SAR Reset state, it shall perform the following:
 - If the value of the ResetCounter field of the ResetTxComplete message is greater than the SARResetCounter, then the SAR may set SARResetCounter to the value of the ResetCounter field.
 - Leave the SAR Reset state.

If a ResetTxComplete message is received for a Stream while that Stream is not in the SAR Reset state, the message shall be ignored.

7.3.4.3.1.1.2.2 Reset procedure for initiating side when it is a SAR receiver

If the side initiating a reset procedure is a SAR receiver for the Stream being reset, then it shall enter the SAR Reset State. Upon entering the SAR Reset state, the SAR receiver shall

- Perform the SAR receiver initialization procedure defined in 7.3.4.3.1.1.1.2, and then send a ResetRxIndication message.
- Ignore all SAR data units received for the Stream being reset while in the SAR Reset state.
- Ignore all Flush messages received for the Stream being reset while in the SAR Reset state.
- When SAR receives a ResetRxComplete message for the Stream being reset, it shall perform the following:
 - If the value of the ResetCounter field of the ResetRxComplete message is greater than the SARResetCounter, then the SAR may set SARResetCounter to the value of the ResetCounter field.
 - Leave the SAR Reset state.

If a ResetRxComplete is received for a Stream while the Stream is not in the SAR Reset state, the message shall be ignored.

7.3.4.3.1.1.2.3 Reset procedure for the responding side when it is a SAR receiver

If the side responding to a reset procedure is a SAR receiver for the Stream being reset, then upon receiving a ResetTxIndication message it shall perform the following:

- If it is not in the SAR Reset state, then perform the SAR receiver initialization procedure defined in 7.3.4.3.1.1.1.2.
- If the value of the ResetCounter field of the ResetTxIndication message is greater than the SARResetCounter, then the SAR may set SARResetCounter to the value of the ResetCounter field.
- Respond with a ResetTxComplete message.

7.3.4.3.1.1.2.4 Reset procedure for the responding side when it is a SAR transmitter

If the side responding to a reset procedure is a SAR transmitter for the Stream being reset, then upon receiving a ResetRxIndication message it shall perform the following:

- If it is not in the SAR Reset state, then perform the SAR transmitter initialization procedure defined in 7.3.4.3.1.1.1.1.
- If the value of the ResetCounter field of the ResetRxIndication message is greater than the SARResetCounter, then the SAR may set SARResetCounter to the value of the ResetCounter field.
- Wait for transmission of all the SAR packets that were being transmitted before receiving ResetRxIndication message to be completed, then respond with a ResetRxComplete message.

7.3.4.3.1.1.2.5 SAR reset message flows

Figure 26 and Figure 27 show message flows for transmitter and receiver initiated SAR reset procedures, respectively.

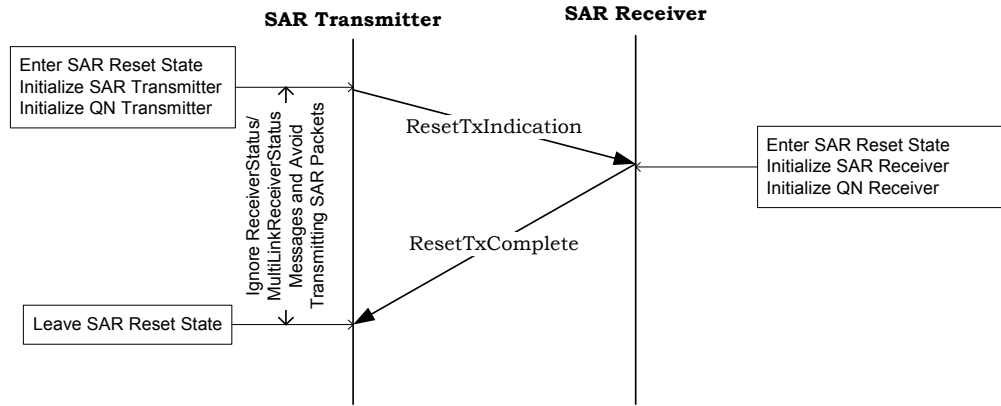


Figure 26—SAR reset procedure initiated by SAR transmitter

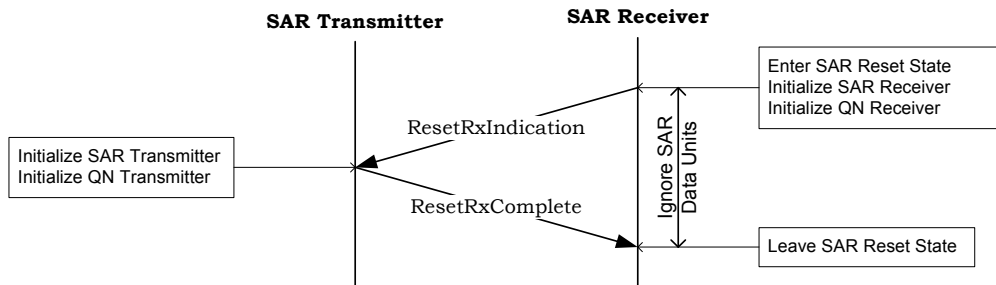


Figure 27—SAR reset procedure initiated by SAR receiver

7.3.4.3.2 Data transfer

SAR is an Ack or Nak-based sub-protocol with a sequence space of SARSequenceLength bits, where SARSequenceLength is indicated by the SequenceLengthFwd attribute for forward Stream and the SequenceLengthRev attribute for reverse Stream.

All operations and comparisons performed on SAR packet sequence numbers shall be carried out in unsigned modulo 2^S arithmetic, where S represents the value of SARSequenceLength. For any SAR sequence number N , the sequence numbers in the range $[N + 1, N + 2^{S-1} - 1]$ shall be considered greater than N and the sequence numbers in the range $[N - 2^{S-1}, N - 1]$ shall be considered smaller than N .

7.3.4.3.2.1 SAR transmit procedures

If the SAR transmitter is the access terminal, then SAR should not start transmission of a new Higher Layer Protocol packet if all of the following conditions are true:

- This is Stream number other than 0x01, 0x02.
- This is not a Reverse Link Serving Route (as indicated by the RLSS public data of the Reverse Control Channel MAC Protocol).

If the SAR transmitter is the access terminal and StreamEarlyOpenRev is set to 0x00, then SAR shall not transmit data units unless all of the following conditions are true:

- SAR has received *RouteControl.RouteOpened* indication.
- SAR has received *RouteControl.LinkInitialized* indication.

The SAR transmitter shall maintain a SARSequenceLength-bit variable $V(S)$ for all SAR data units transmitted using procedures defined in this subclause (see Figure 28). $V(S)$ is the sequence number of the next SAR data unit to be sent on Stream. The sequence number field (SARSEQ) in each new SAR packet transmitted shall be set to $V(S)$, corresponding to the sequence number of the first data unit in the packet. If the data unit is octets, then the sequence number of the i^{th} octet in the packet (with the first octet being octet 0) is implicitly given by $\text{SARSEQ} + i$. $V(S)$ shall be incremented for each data unit contained in the packet.

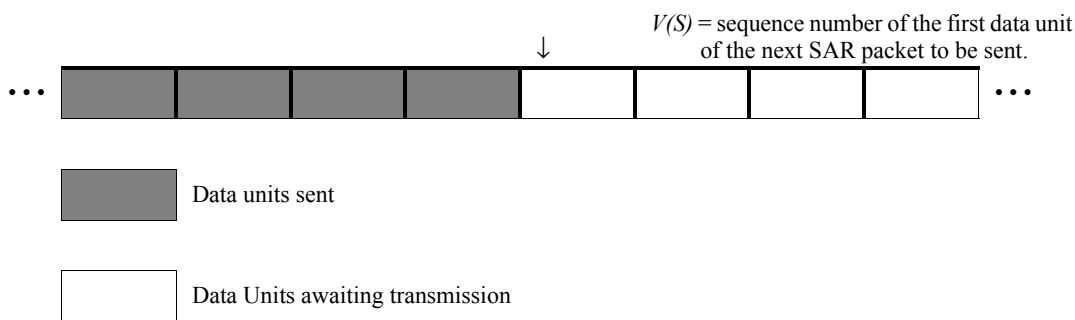


Figure 28—SAR transmit sequence number variable

The SAR transmitter shall maintain a $(48 - \text{SARSequenceLength})$ bit variable SARSequenceRolloverCounter. The SARSequenceRolloverCounter shall be incremented each time $V(S)$ changes from all '1's to all '0's.

The SAR transmitter shall maintain a 8-bit variable SARResetCounter.

The SAR transmitter should allow sufficient time before deleting a SAR packet payload transmitted for the first time.

Upon receiving *AirLinkManagement.IdleInitiated* indication, the SAR transmitter shall perform the following if Stream number corresponding to this protocol is other than 0x00, or 0x03:

- Delete all transmitted packets and any partial packets that are not transmitted.

Upon receiving an ANReceiverStatus message, the SAR transmitter at the access terminal shall transmit the missing data unit(s) (if any) conveyed by the ANReceiverStatus message if all of the following conditions are satisfied:

- The requested data units are available.
- The requested data units have not been retransmitted before in response to an ANReceiverStatus message.
- If the data unit for the Stream is RLP packet payloads and the data unit is not tunneled to the same or a different route, and the payload carrying capacity of the lower layer packet is not smaller than the size of the packet to be retransmitted.

Upon receiving an ATReceiverStatus message, the SAR transmitter at the access network shall transmit the missing data unit(s) (if any) conveyed by the ATReceiverStatus message if all of the following conditions are satisfied:

- The requested data unit is available.

- The requested data unit has not been retransmitted before in response to an ATReceiverStatus message.
- If the data unit for the Stream is RLP packet payloads and the data unit is not tunneled to the same or a different route, the payload carrying capacity of the lower layer packet is not smaller than the size of the packet to be retransmitted.
- One of the following conditions is true:
 - The DelayedNak field corresponding to the missing data units in the ATReceiverStatus message is set to '1', or
 - The DelayedNak field corresponding to the missing data units in the ATReceiverStatus message is not set to '1', the LeadingEdgeValid field corresponding to the missing data units in the ATReceiverStatus message is set to '1', and the missing data unit was transmitted on the QN instance on which data units whose SAR sequence numbers are equal to the LeadingEdge and TrailingEdge fields of the ATReceiverStatus message were transmitted, or
 - The DelayedNak field corresponding to the missing data units in the ATReceiverStatus message is not set to '1', the LeadingEdgeValid field corresponding to the missing data units in the ATReceiverStatus message is set to '0', the missing data unit was transmitted on the QN instance on which data unit whose SAR sequence number is equal to the TrailingEdge field of the ATReceiverStatus message was transmitted, and the missing data unit was transmitted since the last time (before transmission of data unit with sequence number TrailingEdge on the QN instance) that the QN instance became the serving QN instance.

If the SAR transmitter is the access network and if NakAckEnableFwd is set to 0x02, then the access network may determine that transmitted data units have been lost if it does not receive an ATReceiverStatus message acknowledging the receipt of the data units within an implementation-dependent time interval based on the AckTimer and an estimate of the round-trip delay. If the SAR transmitter is the access network and if NakAckEnableFwd is set to 0x02 and the access network determines that transmitted data units were lost, then the access network shall perform the following:

- If the data units have not been previously retransmitted, then access network shall retransmit the data units.
- Otherwise, access network may retransmit the data units. If the access network decides not to retransmit the data units, then the access network shall return a *ReliablePacketDeliveryFailed* indication.

If the SAR transmitter is the access terminal and if NakAckEnableRev is set to 0x02, then the access terminal may determine that transmitted data units have been lost if it does not receive an ANReceiverStatus message acknowledging the receipt of the data units within an implementation-dependent time interval based on the AckTimer and an estimate of the round-trip delay. If the SAR transmitter is the access terminal and if NakAckEnableRev is set to 0x02 and the access terminal determines that transmitted data units were lost, then the access terminal shall perform the following:

- If the data units have not been transmitted MaxTransmissionsRev times, then access terminal shall retransmit the data units.
- Otherwise, access terminal shall discard the data units, and the access terminal shall return a *ReliablePacketDeliveryFailed* indication.

If the SAR transmitter is the access network, and the ATReceiverStatus message includes any sequence number greater than or equal to $V(S)$, SAR shall perform the reset procedures specified in 7.3.4.3.1.1.2.1 for forward Stream. If the SAR transmitter is the access terminal, and the ANReceiverStatus message includes any sequence number greater than or equal to $V(S)$, SAR shall perform the reset procedures specified in 7.3.4.3.1.1.2.1 for reverse Stream. If the ATReceiverStatus or ANReceiverStatus message does not include

any sequence number greater than or equal to $V(S)$ but the requested data units are not available for retransmissions, SAR shall ignore the ATReceiverStatus or ANReceiverStatus message for data units that are not available.

If NakAckEnableFwd or NakAckEnableRev is 0x02 and if SAR initiates reset procedure as specified in 7.3.4.3.1.1.2, then this protocol shall return a *ReliableStreamReset* indication. Upon receiving a *ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed* indication for reverse Stream, the SAR transmitter in the access terminal shall retransmit the requested data unit(s) if and only if all of the following conditions are satisfied:

- PhysicalLayerNak EnableRev attribute is set to 0x01.
- If the data unit for the Stream is RLP packet payloads and the data unit is not tunneled to the same or a different route, the payload carrying capacity of the lower layer packet is not smaller than the size of the packet to be retransmitted.
- The requested data units have not been retransmitted before.
- The requested data units are available.

If NakAckEnableFwd is 0x01, then the transmitter at the access network for this Stream shall meet the following requirements:

- After transmitting a packet that contains first-time data unit(s), the SAR transmitter shall start a SAR flush timer for time FlushTimer, where FlushTimer is a parameter of the TimersFwd attribute.
- If the SAR transmitter sends another packet that contains first-time data unit(s) before the SAR flush timer expires, the SAR transmitter shall reset and restart the timer.
- If the timer expires, the SAR transmitter shall disable the flush timer and the SAR transmitter should perform the following:
 - If NakDelayTimerMultiCarrier or NakDelayTimerSingleCarrier field of the TimersFwd attribute is not set to 0x0000, then send a Flush message containing $V(S) - 1$.
 - Otherwise, either send a Flush message containing $V(S) - 1$ or send a SAR packet that contains at least the data unit with sequence number $V(S) - 1$.

If NakAckEnableRev is 0x01, then the transmitter at the access terminal for this Stream shall meet the following requirements:

- After transmitting a packet that contains first-time data unit(s), the SAR transmitter shall start a SAR flush timer for time FlushTimer, where FlushTimer is a parameter of the TimersRev attribute.
- If the SAR transmitter sends another packet that contains first-time data unit(s) before the SAR flush timer expires, the SAR transmitter shall reset and restart the timer.
- If the timer expires, the SAR transmitter shall disable the flush timer and the SAR transmitter should perform the following:
 - The SAR transmitter should send a SAR packet that contains at least the data unit with sequence number $V(S) - 1$.

If one of the following conditions is true:

- The SAR transmitter is the access network, and the NakAckEnableFwd attribute is not set to 0x02
- The SAR transmitter is the access terminal, and the NakAckEnableRev attribute is not set to 0x02

Then the SAR transmitter should not transmit more than $2^{\text{SARSequenceLength} - 1}$ first-time data units in any AbortTimer interval.

Otherwise, the SAR transmitter should not transmit the data unit $V(S)$ if all of the following conditions are satisfied:

- The data unit $V(S)^{2^{\text{SARSequenceLength} - 1}}$ has not been acknowledged.

- The first transmission of the data unit $V(S)^{-2\text{SARSequenceLength} - 1}$ took place less than AbortTimer interval ago.

7.3.4.3.2.2 SAR receive procedures

The SAR receiver shall maintain two SARSequenceLength-bit variables for receiving, $V(R)$ and $V(N)$ (see Figure 29). $V(R)$ contains the sequence number of the next data unit expected to arrive. $V(N)$ contains the sequence number of the first missing data unit, as described below.

The SAR receiver shall maintain a $(48 - \text{SARSequenceLength})$ bit variable SARSequenceRolloverCounter. The SARSequenceRolloverCounter shall be incremented each time $V(R)$ changes from all '1's to all '0's.

The SAR receiver shall maintain a 8-bit variable SARResetCounter.

In addition, the SAR receiver shall keep track of the status of each data unit in its resequencing buffer indicating whether the data unit was received or not. Use of this status is implied in the procedures that follow.

If the SAR receiver is an access terminal and if the AutoSyncReceiverFwd attribute is set to 0x01, then the access terminal shall perform following procedures:

- When there is no data in resequencing buffer that has not been passed up to the Higher Layer Protocol, the access terminal shall set $V(R)$ and $V(N)$ to NULL.
- If no data unit is received for AbortTimer interval, where AbortTimer is a parameter of the TimersFwd attribute, then the access terminal shall discard all the data in resequencing buffer.

If the SAR receiver is an access network and if the AutoSyncReceiverRev attribute is set to 0x01, then the access network shall perform following procedures:

- When there is no data in resequencing buffer that has not been passed up to the Higher Layer Protocol, the access network shall set $V(R)$ and $V(N)$ to NULL.
- If no data unit is received for AbortTimer interval, where AbortTimer is a parameter of the TimersRev attribute, then the access network shall discard all the data in resequencing buffer.

The SAR receiver informs the SAR transmitter of the status of data units in its receive buffer by sending an ATReceiverStatus or ANReceiverStatus message.¹⁶ The ANReceiverStatus messages shall not convey status of data units with sequence number less than $V(N)$.

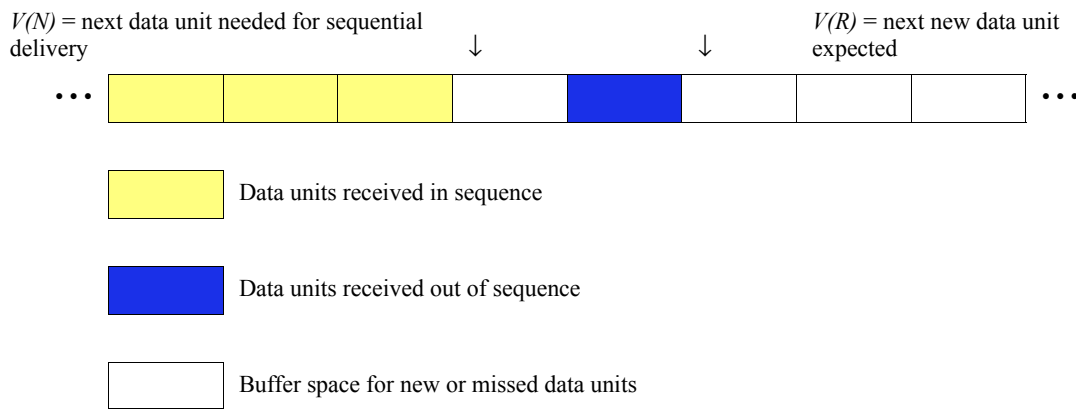


Figure 29—SAR receive sequence number variables

¹⁶Sender needs to consider out of order delivery due to HARQ before sending ATReceiverStatus or ANReceiverStatus message.

If all of the following conditions are true, then the access terminal shall set *NakDelayTimerInUse* to *NakDelayTimerMultiCarrier*, where *NakDelayTimerMultiCarrier* is a parameter of the *TimersFwd* attribute:

- Access terminal has more than one instance of the QN sub-protocol for this Stream.
- Access terminal has more than one *ChannelBand* in its Active Set.

If any of the above conditions are no longer true, then the access terminal shall set *NakDelayTimerInUse* to *NakDelayTimerSingleCarrier* where *NakDelayTimerSingleCarrier* is a parameter of the *TimersFwd* attribute.

The access terminal shall send an *ATReceiverStatus* message upon receiving a *QuickNak.QuickNak* indication if $V(N)$ is not equal to $V(R)$ and if the *NakDelayTimerInUse* is not set to zero.

Upon receiving a Flush message, the SAR receiver shall perform the following if *LastSARSequence* is greater than or equal to $V(R)$:

- If the SAR receiver is an access terminal, the SAR receiver shall send an *ATReceiverStatus* message.
- If the SAR receiver is an access terminal, then SAR shall set a SAR abort timer to *AbortTimer*, where *AbortTimer* is a parameter of the *TimersFwd* attribute, for each missing SAR data unit from $V(R)$ to *LastSARSequence*, inclusive.
- SAR receiver shall set $V(R)$ to *LastSARSequence* + 1.

Upon receiving *AirLinkManagement.IdleInitiated* indication, SAR receiver shall perform the following if Stream number corresponding to this protocol is other than 0x00 or 0x03:

- Discard all the data in resequencing buffer.
- Stop all the SAR abort timers, and Nak delay timers.
- Set $V(N)$ to $V(R)$.

Upon receiving a *TransmitterStatus* message, the SAR receiver shall perform the following for each unavailable data unit conveyed by the message:

- The SAR receiver shall set X to the SAR sequence number of the unavailable data unit.
- If $X = V(N)$, and if $X < V(R)$, then
 - If in-order delivery of Higher Layer Protocol packets is required, then SAR shall pass all complete Higher Layer Protocol packets, that have not been passed to the Higher Layer Protocol, from the beginning of the resequencing buffer upward up to the next missing data unit, to the Higher Layer Protocol. SAR may pass to the Higher Layer Protocol partially received packets with an indication of partial packet delivery.
 - SAR shall set $V(N)$ to the sequence number of the next missing data unit, or to $V(R)$ if there are no remaining missing data units.
 - If SAR abort timer or Nak delay timer is running for this data unit, then SAR shall stop the timer.
- If $X = V(R)$ and if $V(R) = V(N)$, then SAR shall increment $V(N)$ and $V(R)$.

For each received data unit, SAR shall perform the following procedures:

- The SAR receiver shall set X to the SAR sequence number of the received data unit.
- If the SAR receiver is an access terminal and if the *AutoSyncReceiverFwd* attribute is set to 0x01 and if $V(R)$ and $V(N)$ are set to NULL, then the access terminal shall set $V(R)$ and $V(N)$ to X .
- If the SAR receiver is an access network and if the *AutoSyncReceiverRev* attribute is set to 0x01 and if $V(R)$ and $V(N)$ are set to NULL, then the access network shall set $V(R)$ and $V(N)$ to X .

- If the SAR receiver is an access terminal and NakAckEnableFwd is 0x02, then SAR shall send an ATReceiverStatus message reporting received data unit within AckTimer interval of receiving the data unit.
- If the SAR receiver is an access network and NakAckEnableRev is 0x02, then SAR shall send an ANReceiverStatus message within AckTimer interval of receiving the data unit.
- If $X < V(N)$, the data unit shall be discarded as a duplicate.
- If $V(N) \leq X < V(R)$, and the data unit is not already stored in the resequencing buffer nor has been passed up to the Higher Layer Protocol, then
 - SAR shall store the received data unit in the resequencing buffer.
 - If $X = V(N)$, and if in-order delivery of Higher Layer Protocol packets is required, then SAR shall pass all contiguous complete Higher Layer Protocol packets in the resequencing buffer, that have not been passed to the Higher Layer Protocol, from the beginning of the resequencing buffer upward, to the Higher Layer Protocol. SAR shall then set $V(N)$ to (LAST + 1) where LAST is the sequence number of the last contiguous data unit received by the resequencing buffer.
 - If in-order delivery of Higher Layer Protocol packets is not required, then SAR shall pass all complete Higher Layer Protocol packets in the resequencing buffer, that have not been passed to the Higher Layer Protocol, from the beginning of the resequencing buffer upward, to the Higher Layer Protocol. SAR shall then set $V(N)$ to (LAST + 1) where LAST is the sequence number of the last contiguous data unit received by the resequencing buffer.
- If $V(N) < X < V(R)$, and the data unit is already stored in the resequencing buffer or has already been passed up to the Higher Layer Protocol, then the data unit shall be discarded as a duplicate.
- If $X = V(R)$, then
 - If $V(R) = V(N)$, then SAR shall increment $V(N)$ and $V(R)$, shall store the data unit in the resequencing buffer, and shall pass all complete Higher Layer Protocol packets in the resequencing buffer, that have not been passed to the Higher Layer Protocol, from the beginning of the resequencing buffer upward, to the Higher Layer Protocol.
 - If $V(R) \neq V(N)$, SAR shall increment $V(R)$ and shall store the data unit in the resequencing buffer. If in-order delivery of Higher Layer Protocol packets is not required, then SAR shall pass all complete Higher Layer Protocol packets in the resequencing buffer, that have not been passed to the Higher Layer Protocol, from the beginning of the resequencing buffer upward, to the Higher Layer Protocol.
- If $X > V(R)$, then
 - SAR shall store the data unit in the resequencing buffer.
 - If in-order delivery of Higher Layer Protocol packets is not required, then SAR shall pass all complete Higher Layer Protocol packets in the resequencing buffer, that have not been passed to the Higher Layer Protocol, from the beginning of the resequencing buffer upward, to the Higher Layer Protocol.
 - If the SAR receiver is an access network, then SAR shall set a SAR abort timer to AbortTimer, where AbortTimer is a parameter of the TimersRev attribute, for each missing SAR data unit from $V(R)$ to $X - 1$, inclusive. If the SAR receiver is an access terminal, then SAR shall set a SAR abort timer to AbortTimer, where AbortTimer is a parameter of the TimersFwd attribute, for each missing SAR data unit from $V(R)$ to $X - 1$, inclusive.
 - If the SAR receiver is an access terminal, and if the NakAckEnableFwd attribute is not set to 0x00, then SAR shall set a Nak delay timer to NakDelayTimerInUse for each missing SAR data unit from $V(R)$ to $X - 1$, inclusive.
 - If the SAR receiver is an access network, and if the NakAckEnableRev attribute is not set to 0x00, then SAR shall send an ANReceiverStatus message.

- SAR shall set $V(R)$ to $X + 1$.
- If a missing data unit has not been received when its Nak delay timer expires, then the access terminal shall send an ATReceiverStatus message reporting this missing data unit.

For each missing data unit from $V(N)$ upward up to $V(R) - 1$, inclusive, SAR shall perform the following in the order specified, if the SAR abort timer of the missing data unit has expired:

- If in-order delivery of Higher Layer Protocol packets is required, then SAR shall pass all complete Higher Layer Protocol packets, that have not been passed to the Higher Layer Protocol, from the beginning of the resequencing buffer upward up to the next missing data unit, to the Higher Layer Protocol. SAR may pass to the Higher Layer Protocol partially received packets with an indication of partial packet delivery. SAR shall set $V(N)$ to the sequence number of the next missing data unit, or to $V(R)$ if there are no remaining missing data units.

Further recovery is the responsibility of higher layer protocols.

7.3.4.3.3 Message formats

The messages described in this subclause control the function of the SAR. These messages are exchanged between the access terminal and the access network using the Signaling Protocol.

7.3.4.3.3.1 ResetTxIndication

The SAR transmitter in the access terminal or the access network sends the ResetTxIndication message to reset its peer SAR receiver (see Table 45).

Table 45—ResetTxIndication message

Field	Length (bits)
MessageID	8
ResetCounter	8

MessageID The sender shall set this field to 0x00.

ResetCounter The sender shall set this field to value of the SARResetCounter.

Channels	FTC RTC	RLP	Reliable Best Effort
Addressing	Unicast	AuthTag	Required

7.3.4.3.3.2 ResetRxIndication

The SAR receiver in the access terminal or the access network sends the ResetRxIndication message to reset its peer SAR transmitter (see Table 46).

Table 46—ResetRxIndication message

Field	Length (bits)
MessageID	8
ResetCounter	8

MessageID The sender shall set this field to 0x01.

ResetCounter The sender shall set this field to value of the SARResetCounter.

Channels	FTC	RTC

RLP	Reliable	Best Effort

Addressing	Unicast

AuthTag	Required

7.3.4.3.3.3 ResetTxComplete

The SAR receiver in the access terminal or the access network sends the ResetTxComplete message to complete the SAR reset procedure (see Table 47).

Table 47—ResetTxComplete message

Field	Length (bits)
MessageID	8
ResetCounter	8

MessageID The sender shall set this field to 0x02.

ResetCounter The sender shall set this field to value of the SARResetCounter.

Channels	FTC	RTC

RLP	Reliable	Best Effort

Addressing	Unicast

AuthTag	Required

7.3.4.3.3.4 ResetRxComplete

The SAR transmitter in the access terminal or the access network sends the ResetRxComplete message to complete the SAR reset procedure (see Table 48).

Table 48—ResetRxComplete message

Field	Length (bits)
MessageID	8
ResetCounter	8

MessageID The sender shall set this field to 0x03.

ResetCounter The sender shall set this field to value of the SARResetCounter.

Channels	FTC	RTC

RLP	Reliable	Best Effort

Addressing	Unicast

AuthTag	Required

7.3.4.3.3.5 ATReceiverStatus

The access terminal sends the ATReceiverStatus message to request the retransmission of one or more SAR data units (see Table 49).

Table 49—ATReceiverStatus message

Field	Length (bits)
MessageID	8
SARSequenceLength	8
NakReportCount	8

NakReportCount occurrences of the following five fields:

DelayedNak	1
LeadingEdgeValid	1
LeadingEdge	SARSequenceLength
Reserved1	2
TrailingEdge	SARSequenceLength

AckReportCount	8

AckReportCount occurrences of the following five fields:

DuplicateData	1
Reserved2	1
FirstReceivedDataUnit	SARSequenceLength
Reserved3	2
LastReceivedDataUnit	SARSequenceLength

MessageID	The access terminal shall set this field to 0x4.
SARSequenceLength	The access terminal shall set this field to the length of the SAR sequence number as indicated by the SARSequenceLength field of the SequenceLengthFwd attribute for forward Stream.
NakReportCount	The access terminal shall set this field to the number of Nak Report records included in this message. The access terminal shall include NakReportCount occurrences of the following five fields with the message.
DelayedNak	The access terminal shall set this field to '0' if this report is triggered by a <i>QuickNak</i> . <i>QuickNak</i> indication. Otherwise, the access terminal shall set this field to '1'.
LeadingEdgeValid	If DelayedNak is set '1', the access terminal shall set this field to '1'. If DelayedNak is set '0', then

	— If the access terminal has received more than one RLP packet on the corresponding QN instance in the last abort time interval, the access terminal shall set this field to '1' — Otherwise, the access terminal shall set this field to '0'.
LeadingEdge	If LeadingEdgeValid is set to '0', then the access terminal shall set all bits in this field to zero. Otherwise, if this report is triggered by a <i>QuickNak.QuickNak</i> indication, then the access terminal shall include this field and set it as follows: — If this report is triggered by a <i>QuickNak.QuickNak</i> indication, then the access terminal sender shall set this field to the largest SAR sequence number received on the QN instance that triggered this report before the <i>QuickNak.QuickNak</i> indication was received. — If this report is triggered by a Flush message, then the access terminal shall set this field to the value of $V(R) - 1$. — Otherwise, the access terminal shall set this field to one less than the SAR sequence number of the first SAR data unit erased in a sequence of erased data units.
Reserved1	The access network shall ignore this field.
TrailingEdge	If this report is triggered by a <i>QuickNak.QuickNak</i> indication, then the access terminal shall set this field to the SAR sequence number of the first data unit in the payload received along with the <i>QuickNak.QuickNak</i> indication on the QN instance that triggered this report. If this report is triggered by a Flush message, then the access terminal shall set this field to LastSARSequence + 1, where LastSARSequence is the value of the field included in the Flush message. Otherwise, the access terminal shall set this field to one more than the SAR sequence number of the last SAR data unit erased in a sequence of erased data units.
AckReportCount	The access terminal shall set this field to the number of Ack Report records included in this message. If the NakAckEnableFwd is not set to 0x02, then the access terminal shall set this field to 0x00. The access terminal shall include AckReportCount occurrences of the following five fields with the message.
DuplicateData	If the data units included in this Ack Report were discarded as duplicate data units, then the access terminal shall set this field to '1'; otherwise, the access terminal shall set this field to '0'.
Reserved2	The access network shall ignore this field.
FirstReceivedDataUnit	The access terminal shall set this field to the sequence number of the first SAR data unit received in a sequence of received data units.
Reserved3	The access network shall ignore this field.

LastReceivedDataUnit The access terminal shall set this field to the sequence number of the last SAR data unit received in a sequence of received data units.

Channels	RTC	RLP	Best Effort
Addressing	Unicast	AuthTag	Required when key is available

7.3.4.3.3.6 Flush

The access network sends the Flush message to indicate the end of a data burst (see Table 50).

Table 50—Flush message

Field	Length (bits)
MessageID	8
SARSequenceLength	8
LastSARSequence	SARSequenceLength
Padding	0–7 (as needed)

- MessageID

The sender shall set this field to 0x05.
- SARSequenceLength

The sender shall set this field to the size of the SAR sequence number for this Stream in units of bits.
- LastSARSequence

The sender shall set this field to $V(S) - 1$.
- Padding

The sender shall add reserved bits to make the length of the entire message an integer number of octets. The sender shall set these bits to zero. The receiver shall ignore this field.

Channels	FTC	RLP	Best Effort
Addressing	Unicast	AuthTag	Required when key is available

7.3.4.3.3.7 ANReceiverStatus

The access network sends the ANReceiverStatus message to request the retransmission of one or more SAR data units (see Table 51).

Table 51—ANReceiverStatus Message

Field	Length (bits)
MessageID	8
SARSequenceLength	8
NakReportCount	8

NakReportCount occurrences of the following four fields:

Table 51—ANReceiverStatus Message (continued)

Field	Length (bits)
Reserved1	2
LeadingEdge	SARSequenceLength
Reserved2	2
TrailingEdge	SARSequenceLength

AckReportCount	8

AckReportCount occurrences of the following five fields:

DuplicateData	1
Reserved3	1
FirstReceivedDataUnit	SARSequenceLength
Reserved4	2
LastReceivedDataUnit	SARSequenceLength

MessageID	The access network shall set this field to 0x06.
SARSequenceLength	The access network shall set this field to the length of the sequence number as indicated by the SequenceLengthRev attribute for this reverse Stream.
NakReportCount	The access network shall set this field to the number of Nak Report records included in this message. The access network shall include NakReportCount occurrences of the following four fields with the message. Sequences of erased data units shall be listed in the ascending order of SAR sequence number associated with FirstErasedDataUnit.
Reserved1	The access terminal shall ignore this field.
LeadingEdge	The access network shall set this field to one less than the SAR sequence number of the first SAR data unit erased in a sequence of erased data units.
Reserved2	The access terminal shall ignore this field.
TrailingEdge	The access network shall set this field to one more than the SAR sequence number of the last SAR data unit erased in a sequence of erased data units.
AckReportCount	The access network shall set this field to the number of Ack Report records included in this message. If the NakAckEnableRev is not set to 0x02, then the access network shall set this field to 0x00. The access network shall include AckReportCount occurrences of the following five fields with the message.
DuplicateData	If the data units included in this Ack Report were discarded as duplicate data units, then the access network shall set this field to '1'; otherwise, the access network shall set this field to '0'.
Reserved3	The access terminal shall ignore this field.

- FirstReceivedDataUnit The access network shall set this field to the sequence number of the first SAR data unit received in a sequence of received data units.
- Reserved4 The access terminal shall ignore this field.
- LastReceivedDataUnit The access network shall set this field to the sequence number of the last SAR data unit received in a sequence of received data units.

Channels	FTC	RLP	Best Effort
Addressing	Unicast	AuthTag	Required when key is available

7.3.4.3.3.8 TransmitterStatus

The access network or access terminal sends the TransmitterStatus message to indicate that one or more SAR data units are not available at the transmitter (see Table 52).

Table 52—TransmitterStatus message

Field	Length (bits)
MessageID	8
SARSequenceLength	8
DataUnavailableReportCount	8
DataUnavailableReportCount occurrences of the following four fields:	
Reserved1	2
FirstUnavailableDataUnit	SARSequenceLength
Reserved2	2
LastUnavailableDataUnit	SARSequenceLength

- MessageID The access network shall set this field to 0x07.
- SARSequenceLength The access network shall set this field to the length of the sequence number as indicated by the SequenceLengthFwd attribute for this forward Stream. The access terminal shall set this field to the length of the sequence number as indicated by the SequenceLengthRev attribute for this reverse Stream.
- DataUnavailableReportCount
The sender shall set this field to the number of Data Unavailable Report records included in this message. The sender shall include DataUnavailableReportCount occurrences of the following four fields with the message.

Sequences of unavailable data units shall be listed in the ascending order of SAR sequence number associated with FirstUnavailableDataUnit.
- Reserved1 The receiver shall ignore this field.

FirstUnavailableDataUnit The sender shall set this field to the sequence number of the first SAR data unit that is unavailable in a sequence of unavailable data units.

Reserved2 The receiver shall ignore this field.

LastUnavailableDataUnit The sender shall set this field to the sequence number of the last SAR data unit that is unavailable in a sequence of unavailable data units.

Channels	FTC RTC
Addressing	Unicast

RLP	Best Effort
AuthTag	Required when key is available

7.3.4.3.4 Interface to other protocols

7.3.4.3.4.1 Commands

This sub-protocol does not issue any commands.

7.3.4.3.4.2 Indications

This sub-protocol registers to receive the following indications:

- *RouteControl.RouteCreated*
- *RouteControl.RouteOpened*
- *RouteControl.LinkInitialized*
- *AirLinkManagement.IdleInitiated*
- *ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed* (access terminal only) along with parameters indicating the missing data units
- *QuickNak.QuickNak* (access terminal only)

7.3.4.4 Protocol numeric constants

No numeric constants are defined in this sub-protocol.

7.3.5 Quick Nak Sub-Protocol

7.3.5.1 Overview

The Quick Nak Sub-Protocol (QN) is used to detect erased forward link SAR packets. There may be zero or more instances of the QN for each Stream. SAR may deliver SAR packets for transmission to any instance of the QN sub-protocol associated with that Stream. Each instance of the QN sub-protocol detects loss of packets delivered by the SAR transmitter to that QN instance. QN is a sub-protocol associated with the Basic Radio Link Protocol.

7.3.5.2 Primitives and public data

7.3.5.2.1 Commands

This sub-protocol defines the following command:

- *Initialize*

7.3.5.2.2 Return indications

This sub-protocol returns the following indication:

- *QuickNak* (access terminal only)

7.3.5.3 Procedures and messages for the InUse instance of the sub-protocol

This subclause defines the procedures and messages for each instance of QN.

7.3.5.3.1 Procedures

The access network and the access terminal shall have one instance of the QN sub-protocol for each QNGroup. The QNGroup indicates Active Set pilots that share the same QN instance for a given Stream. Information about which Active Set pilots belong to a QNGroup is provided by Scheduler Group public data of the Active Set Management Protocol.

The procedures in this subclause apply to each instance of the QN sub-protocol. If the public data of the Active Set Management Protocol does not provide QNGroup information, then all pilots in the Active Set shall be considered to be part of one QNGroup.

Upon receiving an *ActiveSetManagement.MACResourceUpdated* indication, the access network and the access terminal shall perform the following:

- The access network and the access terminal shall delete QN instances corresponding to QNGroups that have been deleted from the Active Set.
- The access network and access terminal shall create a new QN instance for each new QNGroup in the Active Set.

7.3.5.3.1.1 Initialization

Upon creation of a QN instance, the QN receiver at the access terminal shall set $V(R)_{QN}$ to zero. Upon creation of a QN instance, the QN transmitter at the access network shall set $V(R)_{QN}$ to zero. The QN transmitter at the access terminal shall ignore an *Initialize* command. The QN receiver at the access network shall ignore an *Initialize* command.

Upon receiving an *Initialize* command, the access terminal shall set $V(R)_{QN}$ to zero. Upon receiving an *Initialize* command, the access network shall set $V(R)_{QN}$ to zero.

7.3.5.3.1.2 Data transfer

QN is an erasure detection sub-protocol with a sequence space of 7 bits. All operations and comparisons performed on QN packet sequence numbers (QNSEQ) shall be carried out in unsigned modulo 2^S arithmetic, where S is equal to 7.

7.3.5.3.1.2.1 QN transmit procedures

The SAR sub-protocol associated with each Stream provides SAR packets for transmission to the instance of QN associated with that QNGroup, where QNGroup is the QNGroup associated with the pilot on which the SAR packet is to be transmitted.

The access network shall maintain a 7-bit variable $V(R)_{QN}$ that indicates the sequence number of the next QN packet to be sent using this QN instance. $V(R)_{QN}$ shall be incremented each time a QN packet containing first-time payload is transmitted.

7.3.5.3.1.2.2 QN receive procedures

The access network shall pass the payload of the RLP packet along with the contents of the RLP header to the SAR.

The access terminal shall maintain a 7-bit variable $V(R)_{QN}$ for each QN instance. $V(R)_{QN}$ contains the sequence number of the next QN packet expected to arrive.

If the QN packet does not contain a valid sequence number (QNSEQ), then the access terminal shall pass the payload of the RLP packet to SAR.

In the following, X denotes the sequence number (QNSEQ) of a received RLP packet. For each received RLP packet that contains a valid sequence number (QNSEQ), the access terminal shall perform the following procedures in the order specified:

- QN shall pass the payload of the RLP packet along with the contents of the RLP header to SAR.
- If $X \neq V(R)_{QN}$, then QN shall generate a QuickNak indication.
- QN shall set $V(R)_{QN}$ to $X + 1$.

7.3.5.3.2 Message formats

No messages are defined in this sub-protocol.

7.3.5.4 Interface to other protocols

7.3.5.4.1 Commands

This sub-protocol does not issue any commands.

7.3.5.4.2 Indications

This sub-protocol registers to receive the following indication:

- *ActiveSetManagement.MACResourceUpdated*

7.3.5.5 RLP packet priorities

For a given Stream, the sender shall assign higher priority (lower number) to packets containing retransmitted application traffic than packets containing only first time transmissions.

7.3.5.6 Protocol numeric constants

No numeric constants are defined in this sub-protocol.

7.3.6 Configuration attributes for the Basic Radio Link Protocol

The access terminal shall support default values of all attributes.

7.3.6.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 53. The access terminal and the access network shall use as defaults the values in Table 53 typed in ***bold italics***.

Table 53—Configurable values

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0000	AutoSyncReceiverFwd	Soft/ Dynamic	0x00	Forward link SAR receiver does not automatically synchronize $V(R)$ and $V(N)$. This is default value for Stream number other than 0x00 and 0x03.
			0x01	Forward link SAR receiver automatically synchronizes $V(R)$ and $V(N)$. This is default value for Stream number 0x00 and 0x03.
			All other values	Reserved
0x0001	AutoSyncReceiverRev	Soft/ Dynamic	0x00	Reverse link SAR receiver does not automatically synchronize $V(R)$ and $V(N)$. This is default value for Stream number other than 0x00, and 0x03.
			0x01	Reverse link SAR receiver automatically synchronizes $V(R)$ and $V(N)$.
			All other values	Reserved
0x0002	NakAckEnableFwd	Soft/ Dynamic	0x00	SAR receivers associated with forward Stream do not transmit ATReceiverStatus messages. This is default value for Stream number 0x00, 0x01, 0x03, 0x04, 0x05, and 0x06.
			0x01	SAR receivers associated with forward Stream transmit an ATReceiverStatus message when missing data units are detected. This is default value for Stream number other than 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, and 0x07.
			0x02	SAR receivers associated with forward Stream transmit an ATReceiverStatus message: - When missing data units are detected. - When data units are received, within Acktimer interval of receiving a data unit. This is the default value for Stream number 0x02 and 0x07.
			All other values	Reserved.

Table 53—Configurable values (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0003	NakAckEnableRev	Soft/ Dynamic	0x00	SAR receivers associated with reverse Stream do not transmit ANReceiverStatus messages. This is default value for Stream number 0x01, 0x04, 0x05, and 0x06.
			0x01	SAR receivers associated with reverse Stream transmit an ANReceiverStatus message when missing data units are detected. This is default value for Stream number other than 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, and 0x07.
			0x02	SAR receivers associated with reverse Stream transmit an ANReceiverStatus message: - When missing data units are detected. - When data units are received, within Acktimer interval of receiving a data unit. This is default value for Stream number 0x02 and 0x07.
			All other values	Reserved.
0x0004	PhysicalLayerNakEnableRev	Soft/ Dynamic	0x00	SAR is to ignore <i>ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed</i> indication. This is default value for Stream number other than 0x04, 0x05, and 0x06.
			0x01	SAR is to retransmit data units when a <i>ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed</i> indication is received. This is default value for Stream number 0x04, 0x05, and 0x06.
			All other values	Reserved
0x0005	MaxTransmissionsRev	Soft/ Dynamic	0x00	Reserved
			0x01–0x02	Maximum number of transmissions of SAR data units when Acks are enabled.
			0x03	Maximum number of transmissions of SAR data units when Acks are enabled.
			0x04–0x08	Maximum number of transmissions of SAR data units when Acks are enabled.
			All other values	Reserved

Table 53—Configurable values (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0006	ActivatedFwd	Soft/ Dynamic	0x00	Forward Stream is not activated. This is default value for Stream number other than 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, and 0x08.
			0x01	Forward Stream is activated. This is default value for Stream number 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, and 0x08.
			All other values	Reserved
0x0007	ActivatedRev	Soft/ Dynamic	0x00	Reverse Stream is not activated. This is default value for Stream number other than 0x00, 0x01, 0x02, 0x04, 0x05, 0x06, 0x07, and 0x08.
			0x01	Reverse Stream is activated. This is default value for Stream number 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, and 0x08.
			All other values	Reserved
0x0008	SequenceLengthFwd	Soft/ Dynamic	0x00	SAR of Forward Stream has a 6-bit sequence number.
			0x01	SAR of Forward Stream has a 14-bit sequence number. This is default value for Stream number 0x00, 0x01, and 0x03.
			0x02	SAR of Forward Stream has a 22-bit sequence number. This is default value for Stream number other than 0x00, 0x01, and 0x03.
			0x03	SAR of Forward Stream has a 30-bit sequence number. This value shall not be used unless $\lceil \log_2[R \times (\text{AbortTimer})] \rceil + 1 > 22$ where R is the maximum data rate in units of higher layer octets per second that can be achieved by the lower layers, and abort timer is in units of seconds.
			All other values	Reserved

Table 53—Configurable values (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0009	SequenceLengthRev	Soft/ Dynamic	0x00	SAR of Reverse Stream has a 6-bit sequence number.
			0x01	SAR of Reverse Stream has a 14-bit sequence number. This is default value for Stream number 0x01.
			0x02	SAR of Reverse Stream has a 22-bit sequence number. This is default value for Stream number other than 0x00, 0x01, and 0x03.
			0x03	SAR of Reverse Stream has a 30-bit sequence number. This value shall not be used unless $\lceil \log_2[R \times (\text{AbortTimer})] \rceil + 1 > 22$ where R is the maximum data rate in units of higher layer octets per second that can be achieved by the lower layers, and abort timer is in units of seconds.
			All other values	Reserved
0x000a	DataUnitFwd	Soft/ Dynamic	0x00	Data unit for forward Stream is octets. This is default value for Stream number other than 0x00, 0x01, and 0x03.
			0x01	Data unit for forward Stream is SAR packet payload. This is default value for Stream number 0x00, 0x01, and 0x03.
			All other values	Reserved.
0x000b	DataUnitRev	Soft/ Dynamic	0x00	Data unit for reverse Stream is octets. This is default value for Stream number other than 0x00, 0x01 and 0x03.
			0x01	Data unit for reverse Stream is SAR packet payload. This is default value for Stream number 0x01.
			All other values	Reserved.

Table 53—Configurable values (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x000c	OutOfOrderDeliveryFwd	Soft/ Dynamic	0x00	Forward Stream delivers Higher Layer Protocol payload in-order. This is default value for Stream number other than 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, and 0x06.
			0x01	Forward Stream may deliver Higher Layer Protocol payload out-of-order. This is default value for Stream number 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, and 0x06.
			All other values	Reserved.
0x000d	QNEnableFwd	Soft/ Dynamic	0x00	QN on Forward Stream is not enabled.
			0x01	QN on Forward Stream is enabled.
			All other values	Reserved
0x000e	StreamEarlyOpenRev	Soft/ Dynamic	0x00	Reverse Stream not configured to Open early. This is default value for Stream number other than 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, and 0x07.
			0x01	Reverse Stream configured to Open early. This is default value for Stream number 0x01, 0x02, 0x04, 0x05, 0x06, and 0x07.
			All other values	Reserved
0x000f	TransmitAbortTimerRev	Soft/ Dynamic	0x0000	Maximum time limit for transmission of a higher layer data unit for Stream is not specified.
			0x0001–0x03e8	Maximum time limit for transmission of a higher layer data unit for Stream in units of 5 ms.
			All other values	Reserved

7.3.6.2 Complex attributes

The complex attributes and default values listed in 7.3.6.2.1 through 7.3.6.2.4 are defined (see 15.2.4 for attribute record definition).

7.3.6.2.1 TimersFwd attribute

The sender shall set AttributeID field to 0x8000 (see Table 54).

Table 54—TimersFwd attribute

Field	Length (bits)	Default
AbortTimer	16	Streams 0x00, 0x01, 0x02, 0x03, and 0x07: 0x01f4 All other Streams: 0x00c8
FlushTimer	16	0x00c8
NakDelayTimerMultiCarrier	16	0x0000
NakDelayTimerSingleCarrier	16	0x0032
AckTimer	16	0x0064

AbortTimer The sender shall set this field to the value of the SAR abort timer for this forward Stream in units of milliseconds.

FlushTimer The sender shall set this field to the value of the SAR flush timer for this forward Stream in units of milliseconds. The value of the SAR flush timer shall be less than or equal to that of the corresponding abort timer.

NakDelayTimerMultiCarrier The sender shall set this field to the value of the Nak delay timer, to be used when operating in multi-carrier mode, for this forward Stream in units of milliseconds. The value of the Nak delay timer shall be less than or equal to that of the corresponding abort timer.

NakDelayTimerSingleCarrier The sender shall set this field to the value of the Nak delay timer, to be used when operating in single-carrier mode, for this forward Stream in units of milliseconds. The value of the Nak delay timer shall be less than or equal to that of the corresponding abort timer.

AckTimer The sender shall set this field to the value of the SAR Ack timer for this forward Stream in units of milliseconds.

Commit	Soft

Scope	Dynamic

7.3.6.2.2 TimersRev attribute

The sender shall set AttributeID field to 0x8001 (see Table 55).

Table 55—TimersRev attribute

Field	Length (bits)	Default
AbortTimer	16	Streams 0x01, 0x02, and 0x07: 0x01f4 All other Streams: 0x00c8
FlushTimer	16	0x00c8
AckTimer	16	0x0064

AbortTimer The sender shall set this field to the value of the SAR abort timer for this reverse Stream in units of milliseconds.

FlushTimer The sender shall set this field to the value of the SAR flush timer for this reverse Stream in units of milliseconds. The value of the SAR flush timer shall be less than or equal to that of the corresponding abort timer.

AckTimer The sender shall set this field to the value of the SAR Ack timer for this reverse Stream in units of milliseconds.

Commit	Soft

Scope	Dynamic

7.3.6.2.3 ProtocolParametersFwd attribute

The sender shall set AttributeID field to 0x8002 (see Table 56).

Table 56—ProtocolParametersFwd attribute

Field	Length (bits)	Default for Stream 0x07	Default for Stream 0x08	Default for Stream > 0x08
ProtocolID	8	0x08	0x05	0x00
ProtocolTypeInUse	8 or 16	0xa008	0xa005	0xa000
ProtocolTypeInConfig	8 or 16	0xe008	0xe005	0xe000

ProtocolID The sender shall set this field to an identifier for the Application Sublayer Protocol as specified in 16.1.

ProtocolTypeInUse The sender shall set this field to Protocol Type to be assigned to the InUse instance of Application Layer Protocol. Value of this field shall be in the range of Protocol Types assigned to InUse instances of Application Protocols as specified in 16.1. The sender shall set eight LSBs of this field to the value of ProtocolID field of this attribute.

ProtocolTypeInConfig The sender shall set this field to Protocol Type to be assigned to the InConfiguration instance of Application Layer Protocol. Value of this field shall be in the range of Protocol Types assigned to InConfiguration instances of Application Protocols as specified in 16.1. The sender shall set eight LSBs of this field to the value of ProtocolID field of this attribute.

Commit	Soft

Scope	Dynamic

7.3.6.2.4 ProtocolParametersRev attribute

The sender shall set AttributeID field to 0x8003 (see Table 57).

Table 57—ProtocolParametersRev attribute

Field	Length (bits)	Default for Stream 0x07	Default for Stream 0x08	Default for Stream > 0x08
ProtocolTypeInUse	8 or 16	0xa008	0xa005	0xa000

Table 57—ProtocolParametersRev attribute (continued)

Field	Length (bits)	Default for Stream 0x07	Default for Stream 0x08	Default for Stream > 0x08
ProtocolTypeInConfig	8 or 16	0xe008	0xe005	0xe000
ProtocolID	8	0x08	0x05	0x00

ProtocolID The sender shall set this field to an identifier for the Application Sublayer Protocol as specified in 16.1.

ProtocolTypeInUse The sender shall set this field to Protocol Type to be assigned to the InUse instance of Application Layer Protocol. Value of this field shall be in the range of Protocol Types assigned to InUse instances of Application Protocols as specified in 16.1. The sender shall set eight LSBs of this field to the value of ProtocolID field of this attribute.

ProtocolTypeInConfig The sender shall set this field to Protocol Type to be assigned to the InConfiguration instance of Application Layer Protocol. Value of this field shall be in the range of Protocol Types assigned to InConfiguration instances of Application Protocols as specified in 16.1. The sender shall set eight LSBs of this field to the value of ProtocolID field of this attribute.

Commit	Soft

Scope	Dynamic

7.3.7 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

All configuration attributes and non-attribute data are Session State Information records. This protocol does not define additional parameter records.

7.4 Basic Stream Protocol

7.4.1 Overview

The Basic Stream Protocol provides the ability to multiplex up to 31 streams. Streams listed below are reserved. The other streams can be assigned to applications with different QoS requirements or other applications.

- Stream 0: Broadcast Signaling on forward link and Many-cast Signaling on reverse link
- Stream 1: Best Effort Delivery Signaling
- Stream 2: Reliable Delivery Signaling
- Stream 3: Broadcast Inter Route Tunneling on forward link and Reserved on reverse link
- Stream 4: Best Effort Delivery Inter Route Tunneling¹⁷
- Stream 5: Best Effort Delivery Inter Route Tunneling
- Stream 6: Best Effort Delivery Inter Route Tunneling
- Stream 31: Reserved for future use

¹⁷Multiple Streams are reserved for “Best Effort Delivery Inter Route Tunneling” to simplify scheduling of Inter Route Tunneled packets.

The following Streams are configured for a specific application as listed below, but can be assigned to other applications:

- Stream 7: EAP
- Stream 8: IP with Best Effort QoS

The header added by this protocol is 5 bits in length. If x bits is the length of the payload presented to the Basic Stream Protocol, x shall satisfy $x \bmod 8 = 2$.

7.4.2 Primitives and public data

7.4.2.1 Commands

This protocol does not define any commands.

7.4.2.2 Return indications

This protocol does not return any indications.

7.4.2.3 Local common data

This protocol does not define any local common data.

7.4.2.4 Public data

This protocol shall make the following data public:

- Subtype for this protocol
- All data defined as Static Attribute, Static Non-Attribute Data, and local common data

7.4.3 Protocol data unit

The protocol data unit for this protocol is a Stream Protocol Packet.

7.4.4 Procedures and messages for the InConfiguration instance of the protocol

7.4.4.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

7.4.4.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

The access network shall not initiate negotiation of the MaxStreams attribute.

7.4.4.3 Message formats

This protocol does not define any messages.

7.4.5 Procedures and messages for the InUse instance of the protocol

7.4.5.1 Procedures

This protocol receives packets for transmission from up to 31 different instances of Radio Link Protocols. The protocol adds the Stream header defined in 7.4.5.2 in front of each Radio Link Protocol packet and forwards it for transmission to the Route Protocol.

The protocol receives Stream Protocol packets from the Route Protocol and removes the Stream Protocol header. The protocol packet obtained in this manner is forwarded to the Radio Link Protocol indicated by the Stream field of the Stream Protocol header. For stream 0 on forward link, the receiver shall maintain an independent RLP instance for each sector on each channel from which broadcast messages are received.

Basic Stream Protocol supports up to 31 streams. Each stream carries a Stream (see Radio Link Protocol). Stream can be activated or deactivated. The number of activated Streams on the forward link shall not exceed the value of the MaxActivatedStreamsFwd field of the MaxStreams attribute. The number of activated Streams on the reverse link shall not exceed the value of the MaxActivatedStreamsRev field of the MaxStreams attribute.

The structure of the Stream Protocol packet is shown in Figure 30.

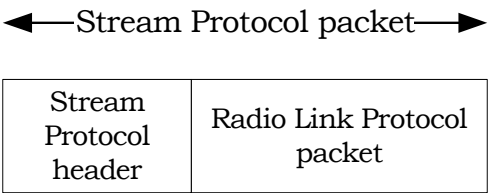


Figure 30—Stream Protocol packet structure

7.4.5.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in 15.1.2.

7.4.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

7.4.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

7.4.5.2 Stream header

The sender adds the following header in front of every Stream Protocol payload:

Stream The sender shall set this field to the stream number associated with the protocol sending the packet following the header (see Table 58).

Table 58—Stream header

Field	Length (bits)
Stream	5

7.4.5.3 Message formats

No messages are defined in this sub-protocol.

7.4.6 Interface to other protocols

7.4.6.1 Commands

This protocol does not issue any commands.

7.4.6.2 Indications

This protocol does not register to receive any indications.

7.4.7 Configuration attributes for Basic Stream Protocol

The attributes and default values listed in 7.4.7.1 and 7.4.7.2 are defined (see Clause 15 for attribute record definition).

7.4.7.1 Simple attributes

This protocol does not define any simple attributes

7.4.7.2 Complex attributes

This protocol defines the complex attributes listed in 7.4.7.2.1.

7.4.7.2.1 MaxStreams attribute

The sender shall set AttributeID field to 0x8000 (see Table 59).

Table 59—MaxStreams attribute

Field	Length (bits)	Default
MaxActivatedStreamsFwd	8	0x0b
MaxActivatedStreamsRev	8	0x0b

MaxActivatedStreamsFwd The sender shall set this field to indicate the maximum number of simultaneous activated forward Streams supported. The value shall be in the range of 0x0b to 0x1f, inclusive.

MaxActivatedStreamsRev The sender shall set this field to indicate the maximum number of simultaneous activated reverse Streams supported. The value shall be in the range of 0x0b to 0x1f, inclusive.

Commit	Hard

Scope	Dynamic

7.4.8 Protocol numeric constants

Protocol numeric constants are listed in Table 60.

Table 60—Protocol numeric constants

Constant	Meaning	Value
N _{STRType}	Type field for this protocol.	See 16.1
N _{STRBasic}	Subtype field for this protocol	See 16.1

7.4.9 Non-attribute data

This protocol does not define any non-attribute data.

7.4.10 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of only the configuration attributes of this protocol.

7.5 Basic Route Protocol

7.5.1 Overview

The Basic Route Protocol performs the following functions:

- Adds Route Protocol Header to convey the parameters used to identify the Protocol Stack associated with this Route during Route initialization.
- Determines whether Route Protocol packets for transmission should be delivered to the Packet Consolidation Protocol of this Route or the Inter-Route Tunneling Protocol of this or another Route.

A Route consists of an InUse protocol stack associated with an access network.

7.5.2 Primitives, local common data, and public data

7.5.2.1 Commands

This protocol does not define any commands.

7.5.2.2 Return indications

This protocol returns the following indications:

- *ATIReceived* (ATIType, ATI, RouteStatus)
- *PersonalityFailure*
- *RouteExists*
- *RouteReopen*
- *UATIFailed*

7.5.2.3 Local common data

This protocol does not define any local common data.

7.5.2.4 Public data

- This protocol shall make the following data public:
- Subtype for this protocol.
 - All data defined as Static Attribute, Static Non-Attribute Data, and local common data.

7.5.3 Protocol data unit

The protocol data unit for this protocol is a Route Protocol Packet (see Figure 31). A Route Protocol Packet consists of Route Protocol Payload and a Route Protocol Header.

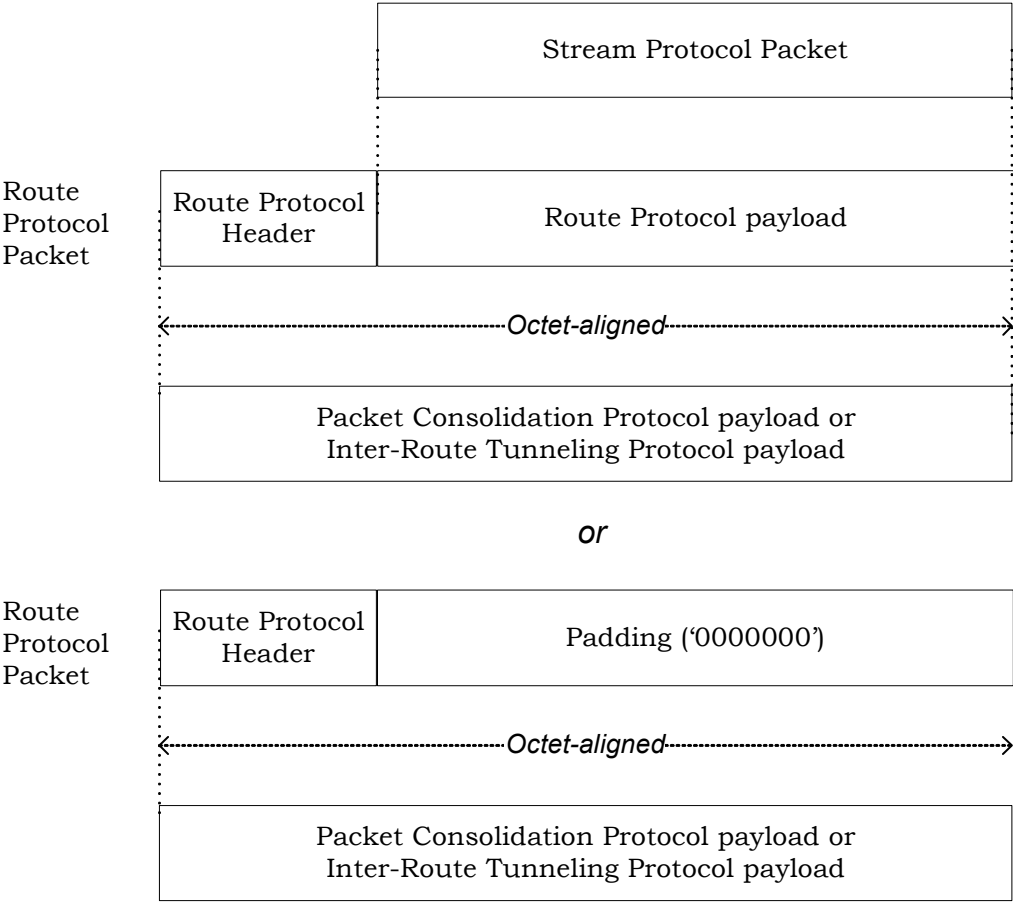
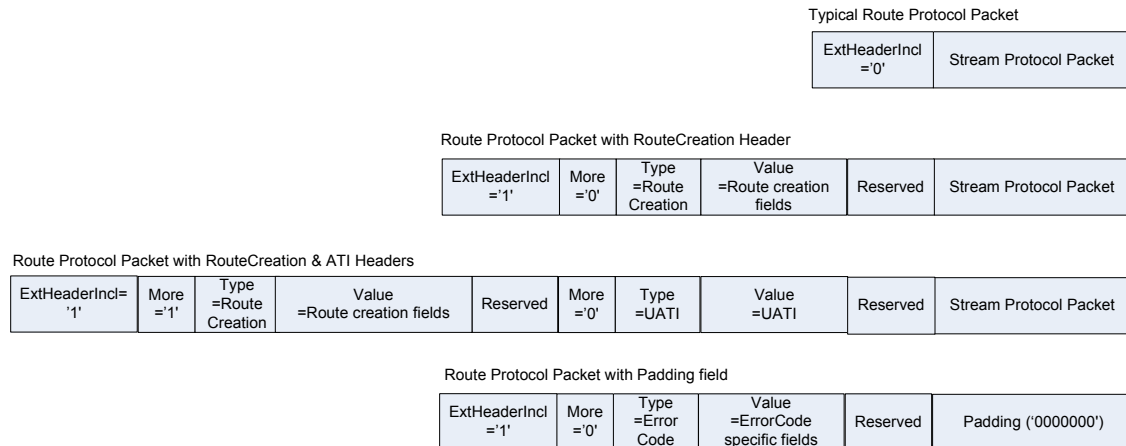


Figure 31—Basic Route Protocol packet

Some example Route Protocol Header configurations are shown in Figure 32.

**Figure 32—Basic Route Protocol Header examples****7.5.4 Procedures and messages for the InConfiguration instance of the protocol****7.5.4.1 Procedures**

This protocol uses the Session Control Protocol to perform negotiation of attribute values.

7.5.4.2 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

7.5.4.3 Message formats

This protocol does not define any messages.

7.5.5 Procedures and messages for the InUse instance of the protocol**7.5.5.1 Procedures****7.5.5.1.1 Protocol initialization**

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in 15.1.2.

7.5.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

7.5.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

7.5.5.1.4 Route Protocol Header

The Route Protocol Header has the format shown in Table 61.

Table 61—Route Protocol Header

Field	Length (bits)
ExtendedHeaderIncluded	1
Zero or more instances of the below record:	
HeaderElement	8n where n is an integer > 0

HeaderElement format:

MoreHeader	1
HeaderType	3
HeaderTypeSpecificFields	Variable
Padding	0–7 (as needed)

If HeaderType= '000', HeaderTypeSpecificFields are as follows:

PSIIncluded	1
IPSI	0 or 4
PersonalityIndex	0 or 4
PSI	0 or 16
RouteID	7
DeleteOldRoutes	1

If HeaderType= '001', HeaderTypeSpecificFields are as follows:

ATIType	2
ATI	128

If HeaderType= '010', HeaderTypeSpecificFields are as follows:

ErrorCode	4
RouteID	0 or 7
CRCErrDetectPattern	0 or 16

ExtendedHeaderIncluded The sender shall set this field to '1' if one or more HeaderElement records are included; otherwise, the sender shall set this field to '0'.

If the ExtendedHeaderIncluded field is set to '1', the sender shall include one or more occurrences of the following HeaderElement record:

MoreHeader The sender shall set this field to '1' if there is another HeaderElement record following this HeaderElement record; otherwise, the sender shall set this field to '0'.

HeaderType The sender shall include this field and set it as specified in Table 62 to indicate the type of HeaderElement record.

Table 62—HeaderType values

HeaderType (binary)	Type of HeaderElement record
000	RouteCreation (Reverse link only)
001	ATI (Forward link and Reverse link)
010	ErrorCode (Forward link only)
Other values	Reserved

If the HeaderType field is set to '000', then the access terminal shall include the following variable-length record:

PSIIncluded	The access terminal shall set this field to '1' if the PSI field is included in this header. The access terminal shall set this field to '0' if the IPSI and PersonalityIndex fields are included in this header.
IPSI	The access terminal shall omit this field if the PSIIncluded field is set to '1'; otherwise, the access terminal shall include this field and set it as follows: The access terminal shall set this field to the InitialProtocolSetIdentifier corresponding to the Personality selected for this Route. When the personality selected for this Route corresponds to a stored personality, this field shall be set to the InitialProtocolSetIdentifier associated with the stored personality.
PersonalityIndex	The access terminal shall omit this field if the PSIIncluded field is set to '1'; otherwise, the access terminal shall include this field and set it as follows: The access terminal shall set this field to the PersonalityIndex of the personality selected for this Route. When the Personality selected for this Route does not correspond to a stored personality (i.e., an IPSI), the access terminal shall set this field to '1111'.
PSI	The access terminal shall omit this field if the PSIIncluded field is set to '0'; otherwise, the access terminal shall include this field and set it as follows: The access terminal shall set this field to the ProtocolSetIdentifier corresponding to the Personality selected for this Route.
RouteID	The access terminal shall set this field to the RouteID assigned to this Route. It shall be set the seven LSBs of the RouteCounter used in the corresponding Route open request.
DeleteOldRoutes	The access terminal shall set this field to '1' if the access network is required to delete any existing Routes to this access terminal (if any).

If the HeaderType field is set to '001', the sender shall include the following two-field record:

ATIType	The sender shall set this field as specified in Table 63.

Table 63—ATType field encoding

ATType	ATType Description	ATI Length (bits)
'00'	Reserved	N/A
'01'	Reserved	N/A
'10'	Unicast ATI (UATI)	128
'11'	Random ATI (RATI)	128

ATI If the ATType field is set to '10', the sender shall set this field to the UATI (i.e., currentATI public data of the Route Control Protocol) corresponding to this access terminal.

If the ATType field is set to '11', the sender shall set this field to the RATI (i.e., currentATI public data of the Route Control Protocol) corresponding to this access terminal.

If the HeaderType field is set to '010', the access network shall include the following variable-length record:

ErrorCode The access network shall set this field as specified in Table 64 to indicate the error code.

Table 64—ErrorCode values

ErrorCode (binary)	Error reason
0000	Personality not supported
0001	Route exists
0010	Close session
0011	Route does not exist
0100	UATI Failed
Other values	Reserved

RouteID The access network shall omit this field if the ErrorCode field is not set to '0001'; otherwise, the access network shall include this field and set it to the RouteID corresponding the existing Route.

CRCErrDetectPattern The access network shall omit this field if the ErrorCode field is not set to '0010'; otherwise, the access network shall include this field and set it to '1010101010101010'.

Padding The sender shall include zero to seven bits to make this HeaderElement record octet-aligned. The sender shall set these bits to '0'. The receiver shall ignore these bits.

7.5.5.1.5 Transmission procedures

7.5.5.1.5.1 Access terminal procedure

If the Basic Route Protocol receives a Stream Protocol Packet for transmission or wants to send a Route Protocol Packet with no payload,¹⁸ the Basic Route Protocol shall perform the following:

- The Basic Route Protocol shall add a Route Protocol Header to create a Route Protocol Packet as follows:
 - The Basic Route Protocol shall determine if one or more HeaderElement records need to be added as follows:
 - + A HeaderElement record of HeaderType of '000' (i.e., RouteCreation header) shall be added if and only if all of the following conditions are satisfied:
 - ♦ State public data of the Route Control Protocol is in WaitingToOpen state.
 - ♦ The access terminal has not received any packets from the access network on this Route.
 - + If a HeaderElement record of HeaderType of '000' (i.e., RouteCreation header) is being added, the access terminal shall perform the following: if there is a pending personality switch request from this access network, the access terminal shall set the PersonalityIndex field of the header to the value of PendingPersonalityIndex public data of the Session Control Protocol.
 - + A HeaderElement record of HeaderType of '001' (i.e., ATI header) shall be added if and only if all of the following conditions are satisfied:
 - ♦ The ProtocolState public data of the Connected State Protocol is set to BindATI.
 - ♦ The access terminal has not received a Route Protocol header from the access network containing an ATI header identical to the one sent by the access terminal.
 - + If a HeaderElement record of HeaderType of '001' (i.e., ATI header) is being added, the access terminal shall perform the following:
 - ♦ If the currentATI public data of the Route Control Protocol is set to UATI, the access terminal shall set the ATIType field to '10' and shall set the ATI field to the currentATI public data of the Route Control Protocol.
 - ♦ If the currentATI public data of the Route Control Protocol is set to RATI, the access terminal shall set the ATIType field to '11' and shall set the ATI field to the currentATI public data of the Route Control Protocol.
 - + If more than one Packet Consolidation Protocol Packet will be carried in a single MAC packet, and the HeaderElements are the same in all Packet Consolidation Protocol packets, the Basic Route Protocol should omit the HeaderElement records in all Packet Consolidation Protocol Packets except the first one included in the MAC packet.
 - + If there is no need to add any HeaderElement records, the Basic Route Protocol shall set the ExtendedHeaderIncluded field to '0'; otherwise, the Basic Route Protocol shall set the ExtendedHeaderIncluded field to '1' and add the required number of HeaderElement records.
 - If this corresponds to a Route Protocol Packet with no payload, the Basic Route Protocol shall set the Padding field to '0000000'.
 - If there is currently no Route that is the Reverse Link Serving Route (as indicated by the RLSS public data of the Reverse Control Channel MAC Protocol), the Route Protocol shall issue an Airlink-

¹⁸At the access terminal the Route Protocol sends a Packet with no payload when it has to provide a payload to the MAC Layer and no Stream Protocol Packet is available.

Management.OpenConnection command; otherwise, the Route Protocol shall perform the following:

- If this Route is the Reverse Link Serving Route (as indicated by the RLSS public data of the Reverse Control Channel MAC Protocol), the Basic Route Protocol should deliver the Route Protocol Packet to the Packet Consolidation Protocol of its Route but may deliver the Route Protocol Packet to the Inter-Route Tunneling Protocol of its Route.
- Otherwise, the Basic Route Protocol shall deliver the Route Protocol Packet to the Inter-Route Tunneling Protocol of the Reverse Link Serving Route (as indicated by the RLSS public data of the Reverse Control Channel MAC Protocol).

7.5.5.1.5.2 Access network procedures

If the Basic Route Protocol receives a Stream Protocol Packet for transmission or wants to send a Route Protocol Packet with no payload, the Basic Route Protocol shall perform the following:

- The Basic Route Protocol shall add a Route Protocol Header to create a Route Protocol Packet as follows:
 - The Basic Route Protocol shall determine if one or more HeaderElement records need to be added as follows:
 - + A HeaderElement record of HeaderType of '010' (i.e., ErrorCode header) shall be included as follows:
 - ♦ If the access network receives a RouteCreation header from the access terminal, and the personality proposed by the access terminal is not supported by the access network, the access network shall set the ErrorCode field to '0000'.
 - ♦ If the access network receives a RouteCreation header from the access terminal with DeleteOldRoutes set to '0', a Route already exist for this access terminal, and the State public data of the Route Control Protocol is not set to WaitingTo-Close state, the access network shall set the ErrorCode field to '0001'.
 - ♦ ErrorCode field shall be set to '0010' if the access network is requesting to close the session.
 - + A HeaderElement record of HeaderType of '010' (i.e., ErrorCode header) shall be included as follows:
 - ♦ If the access network receives a HeaderElement record of HeaderType of '001' (i.e., ATI header) from the access terminal but without a HeaderElement record of HeaderType of '000' (i.e., RouteCreation header), then the access network shall set the access network shall set the ErrorCode field to '0011'.
 - + A HeaderElement record of HeaderType of '001' (i.e., ATI header) shall be included according to the following requirements:
 - ♦ It shall be included if and only if ProtocolState public data of the Connected State Protocol is set to BindATI.
 - ♦ The access network shall set the ATIType and ATI fields to the corresponding values received from the access terminal.
 - If there is no need to add any HeaderElement records based on the above rules, the Basic Route Protocol shall set the ExtendedHeaderIncluded field to '0'; otherwise, the Basic Route Protocol shall set the ExtendedHeaderIncluded field to '1' and add the required number of HeaderElement records.
 - If this corresponds to a Route Protocol Packet with no payload, the Basic Route Protocol shall set the Padding field to '0000000'.
- The Basic Route Protocol shall perform the following:

- If this Route is the Forward Link Serving Route(as indicated by the FLSS public data of the Reverse Control Channel MAC Protocol), the Basic Route Protocol should deliver the Route Protocol Packet to the Packet Consolidation Protocol of its Route but may deliver the Route Protocol Packet to the Inter-Route Tunneling Protocol of its Route.
- Otherwise, the Basic Route Protocol shall deliver the Route Protocol Packet to the Inter-Route Tunneling Protocol of the Forward Link Serving Route (as indicated by the FLSS public data of the Reverse Control Channel MAC Protocol).

7.5.5.1.6 Reception procedures

7.5.5.1.6.1 Access terminal procedures

If the Basic Route Protocol receives a Route Protocol Packet from the Packet Consolidation Protocol of its Route or from the Inter-Route Tunneling Protocol of its Route or another Route, the Basic Route Protocol shall perform the following:

- The Basic Route Protocol shall remove the Route Protocol Header present to produce a Stream Protocol Packet or a Padding field as follows:
 - a) If a HeaderElement record of HeaderType of '010' (i.e., ErrorCode header) is present with ErrorCode field set to '0000', the Basic Route Protocol shall return a PersonalityFailure indication.
 - b) If a HeaderElement record of HeaderType of '010' (i.e., ErrorCode header) is present with ErrorCode field set to '0001', the Basic Route Protocol shall return a RouteExists indication.
 - c) If a HeaderElement record of HeaderType of '010' (i.e., ErrorCode header) is present with ErrorCode field set to '0010' and the CRCErrorDetectPattern field is set to '1010101010101010', the Basic Route Protocol shall issue a SessionControl.Deactivate command.
 - d) If a HeaderElement record of HeaderType of '010' (i.e., ErrorCode header) is present with ErrorCode field set to '0011', the access terminal should create a Route to the access network that sent this error code.
 - e) If a HeaderElement record of HeaderType of '010' (i.e., ErrorCode header) is present with ErrorCode field set to '0100', the Basic Route Protocol shall return a UATIFailed indication.
 - f) The Basic Route Protocol shall return a ATIReceived (ATIType, ATI, RouteStatus) indication, where the arguments are set as follows:
 - 1) If a HeaderElement record of HeaderType of '001' (i.e., ATI header) is present, ATIType and ATI arguments shall be set to the corresponding values received in the header.
 - 2) If a HeaderElement record of HeaderType of '001' (i.e., ATI header) is not present, ATIType and ATI arguments shall be set to NULL.
 - 3) If a HeaderElement record of HeaderType of '001' (i.e., ATI header) is not present or if a HeaderElement record of HeaderType of '010' was present, RouteStatus shall be set to 0x1; otherwise, RouteStatus shall be set to 0x0.
- If the Route Protocol Packet consists of a Padding field of '0000000' and not a Stream Protocol Payload, the Basic Route Protocol shall discard the Padding field; otherwise, the Basic Route Protocol shall deliver the Stream Protocol Packet to the Stream Protocol of its Route.

7.5.5.1.6.2 Access network procedures

If the Basic Route Protocol receives a Route Protocol Packet from the Packet Consolidation Protocol of its Route or from the Inter-Route Tunneling Protocol of its Route or another Route, the Basic Route Protocol shall perform the following:

- The Basic Route Protocol shall remove the Route Protocol Header present to produce a Stream Protocol Packet or a Padding field as follows:
 - a) If a HeaderElement record of HeaderType of '001' (i.e., ATI header) is present, the Basic Route Protocol shall return a ATIReceived (ATIType, ATI) indication with ATI set to the received value of ATI.
 - b) If a HeaderElement record of HeaderType of '000' (i.e., RouteCreation header) is present, the access network shall perform the following: the access network shall ignore this header if State public data of the Route Control Protocol is not equal to WaitingToOpen or WaitingToClose; if the header is not ignored, the access network shall perform the following:
 - 1) If the DeleteOldRoutes field is set to '1', the Basic Route Protocol shall issue a DeleteOldRoutes command.
 - 2) If State public data of the Route Control Protocol is set to WaitingToClose, the Route Protocol shall return a RouteReopen indication.
- If the Route Protocol Packet consists of a Padding field of '0000000' and not a Stream Protocol Payload, the Basic Route Protocol shall discard the Padding field; otherwise, the Basic Route Protocol shall deliver the Stream Protocol Packet to the Stream Protocol of its Route.

7.5.5.2 Message formats

This protocol does not define any messages.

7.5.6 Interface to other protocols

7.5.6.1 Commands sent

This protocol issues the following commands:

- *AirlinkManagement.OpenConnection*
- *SessionControl.Deactivate*
- *RouteControl.DeleteOldRoutes*

7.5.6.2 Indications

This protocol does not register to receive any indications.

7.5.7 Configuration attributes

This protocol does not define any simple attributes or complex attributes.

7.5.8 Non-attribute data

This protocol does not define any non-attribute data.

7.5.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records. All configuration attributes and non-attribute data are Session State Information records. This protocol does not define additional parameter records.

8. Lower MAC Sublayer

8.1 General

8.1.1 Overview

This clause defines the normative behavior of the Lower MAC layer by describing the precise processing rules for each of the Lower MAC layer entities. This clause also describes the overall structure of the Lower MAC layer, defines the entity and service model that is used within this standard, and specifies the entities that comprise the Lower MAC layer. This clause consists of the following subclauses:

- **Basic Packet Consolidation Protocol.** This protocol provides the procedures and messages to carry multiple upper layer payloads in one Lower MAC Layer packet.
- **Basic Superframe Preamble MAC Protocol.** This subclause provides the procedures and messages for the operation of the channels transmitted on the superframe preamble.
- **Basic Access Channel MAC Protocol.** This subclause provides the procedures and messages for the operation of the reverse access channel.
- **Basic Forward Link Control Segment MAC Protocol.** This subclause provides the procedures and messages for the operation of the channels transmitted on the forward link control segment.
- **Basic Forward Traffic Channel MAC Protocol.** This subclause provides the procedures and messages for the operation of the forward data channel.
- **Basic Reverse Control Channel MAC Protocol.** This subclause provides the procedures and messages for the operation of all the reverse link control channels except the reverse access channel.
- **Basic Reverse Traffic Channel MAC Protocol.** This subclause provides the procedures and messages for the operation of the reverse data channels.

8.1.2 Power density

The Lower MAC Layer computes the power settings of Physical Layer channels and passes these settings to the Physical Layer for use. In some cases, “power density” is used to communicate transmission power levels defined as follows:

Power density	The expected energy per modulation symbol ¹⁹ where “modulation symbol” refers to frequency domain (pre-IFFT) symbols, as defined in the Physical Layer specification. The “expected” power is used to cover the case of transmissions using non-constant modulus signal constellations. In all cases, the density shall be computed according to the parameters of the modulation symbols in the particular channel being specified.

8.1.3 Data encapsulation

In the transmit direction at the access network, the Lower MAC Layer receives Route packets, consolidates zero or more Route packets into a Consolidated Packet using the Packet Consolidation Protocol, invokes the Forward Traffic Channel MAC to add layer-related headers and trailers, and forwards the resulting packet to the Physical Layer for transmission on the F-DCH.

In the transmit direction at the access terminal, the Lower MAC Layer receives Route packets, consolidates zero or more Route packets into a Consolidated Packet using the Packet Consolidation Protocol, invokes the

¹⁹Note that this definition of power density is not true in the strictest sense because of cyclic prefix and windowing overheads. However, it provides a convenient method of communicating power levels to the Physical Layer Protocol. The construction of transmitted waveforms as a function of power densities is specified by the Physical Layer Protocol.

Reverse Traffic Channel MAC to add layer-related headers and trailers, and forwards the resulting packet to the Physical Layer for transmission on the R-ODCH or R-CDCH.

In the receive direction at the access terminal, the Lower MAC Layer receives Lower MAC packets from the Physical Layer F-DCH, uses the Forward Traffic Channel MAC to remove layer-related headers and trailers, uses the Packet Consolidation Protocol to de-multiplex Route packets, and forwards the contained Route packets to the Route Protocol.

In the receive direction at the access network, the Lower MAC Layer receives Lower MAC packets from the Physical Layer R-ODCH or R-CDCH, uses the Reverse Traffic Channel MAC to remove layer-related headers and trailers, uses the Packet Consolidation Protocol to de-multiplex Route packets, and forwards the contained Route packets to the Route Protocol.

Figure 33 illustrates the relationship between Route packets, Consolidated packets, Lower MAC packets, and Physical Layer packets. The content and formulation of Lower MAC headers and trailers is specified for each protocol in this clause.

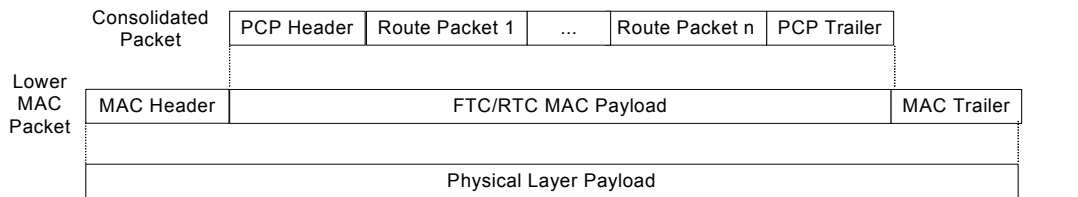


Figure 33—Packet encapsulation

8.1.4 Superframe structure for the Basic Physical Layer

A superframe is the fundamental unit of transmission on both forward and reverse links. A forward link superframe consists of a superframe preamble followed by $N_{\text{FLPHYFrames}}$ FL PHY frames. A reverse link superframe consists of $N_{\text{RLPHYFrames}}$ RL PHY frames. Frequency Division Duplex (FDD) and Time Division Duplex (TDD) superframe timing are described in the following subclauses.

8.1.4.1 FDD superframe structure

The contents of this subclause refers to an FDD system; this will be specified by the value of the Overhead Messages Protocol (OMP) variable `DuplexingMode = FDD`. If `DuplexingMode = FDD`, then $N_{\text{FLPHYFrames}} = N_{\text{RLPHYFrames}} = 25$.

The first RL PHY frame in each superframe is lengthened by the length of the superframe preamble to ensure that superframes are time-aligned between forward and reverse links.²⁰

Note that there will be a guard time between two adjacent PHY frames (on both FL and RL) if half-duplex operation is enabled. The guard time is not relevant for the specification of Lower MAC layer protocols and will be described in the Physical Layer.

Each superframe shall be uniquely identified by a superframe index that is incremented every superframe. The superframe index is related to the system time as described in the Physical Layer. Furthermore, each FL PHY frame and each RL PHY frame shall be uniquely identified by an FL PHY frame index and an RL PHY frame index respectively. The FL PHY frame index of the i^{th} FL PHY frame in the j^{th} superframe is given by

²⁰The longer RL PHY frame results in a larger number of OFDM symbols. These OFDM symbols are used to carry extra redundant bits for the same HARQ transmission as described in the Physical Layer.

$j \times N_{\text{FLPHYFrames}} + i$, where $i = 0, 1, \dots, N_{\text{FLPHYFrames}} - 1$, and $j = 0$. The RL PHY frame index of the i^{th} RL PHY frame in the j^{th} superframe is given by $j \times N_{\text{RLPHYFrames}} + i$, where $i = 0, 1, \dots, N_{\text{RLPHYFrames}} - 1$, and $j \geq 0$. The above indexing scheme for PHY frames is used throughout the MAC layer specification. However, a different indexing scheme may be used in other layers of this standard.

Henceforth, a PHY frame (FL or RL) with PHY frame index k is referred to as (FL or RL) PHY frame k . The exact timing details are provided in the Physical Layer. The structure of forward link and reverse link superframes is shown in Figure 34.

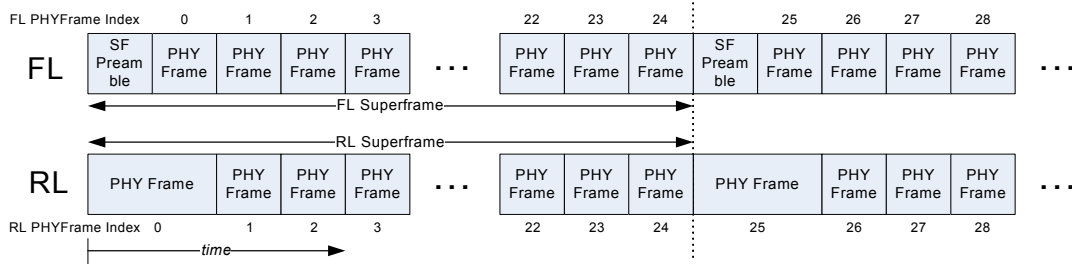


Figure 34—FDD superframe timing

8.1.4.2 TDD superframe structure

The contents of this subclause refers to a TDD system; this will be specified by the value of the OMP variable `DuplexingMode = TDD44` or `DuplexingMode = TDD63`. The two values specify the transmission scheme used. TDD mode has two associated variables called $N_{\text{FL\_BURST}}$ and $N_{\text{RL\_BURST}}$ that determine FL and RL partitioning. The variables $N_{\text{FL\_BURST}}$ and $N_{\text{RL\_BURST}}$ are determined based on the value of the `DuplexingMode` variable. When `DuplexingMode = TDD44`, the system operates in TDD 4:4 partitioning mode with $N_{\text{FL\_BURST}} = N_{\text{RL\_BURST}} = 4$; when `DuplexingMode = TDD63`, the system operates in TDD 6:3 partitioning mode with $N_{\text{FL\_BURST}} = 6$ and $N_{\text{RL\_BURST}} = 3$.

A TDD forward link superframe consists of a superframe preamble and $N_{\text{FLPHYFrames}}$ FL PHY frames, and a TDD reverse link superframe consists of $N_{\text{RLPHYFrames}}$ RL PHY frames where $N_{\text{FLPHYFrames}}$ is the number of FL PHY frames per FL superframe, and $N_{\text{RLPHYFrames}}$ is the number of RL PHY frames per RL superframe. Denote by N_{SF} the total number of PHY frames in one superframe. If `DuplexingMode = TDD44`, then $N_{\text{SF}} = 24$ and $N_{\text{FLPHYFrames}} = N_{\text{RLPHYFrames}} = 12$. If `DuplexingMode = TDD63` then $N_{\text{SF}} = 27$, $N_{\text{FLPHYFrames}} = 18$ and $N_{\text{RLPHYFrames}} = 9$.

In the TDD mode, FL and RL transmissions alternate in time. Excluding the superframe preamble at the beginning of each FL superframe, the transmissions alternate between FL PHY frames and RL PHY frames throughout the superframe. An appropriate amount of guard interval is inserted between each FL-RL transition. The guard time is not relevant for the specification of MAC layer protocols and will be described in the Physical Layer.

Each superframe shall be uniquely identified by a superframe index that is incremented every superframe. The superframe index is related to the system time as described in the Physical Layer. Furthermore, each FL PHY frame and each RL PHY frame shall be uniquely identified by an FL PHY frame index and an RL PHY frame index respectively.

The FL PHY frame index of the i^{th} FL PHY frame in the j^{th} superframe is given by $j \times N_{\text{FLPHYFrames}} + i$, where $i = 0, 1, \dots, N_{\text{FLPHYFrames}} - 1$, and $j \geq 0$. The RL PHY frame index of the i^{th} RL PHY frame in the j^{th} superframe is given by $j \times N_{\text{RLPHYFrames}} + i$, where $i = 0, 1, \dots, N_{\text{RLPHYFrames}} - 1$, and $j \geq 0$. A PHY frame (FL or RL) with PHY frame index k is referred to as (FL or RL) PHY frame k . The exact timing details are provided in the Physical Layer.

The previous indexing scheme for PHY frames is used throughout the MAC layer. However, a different indexing scheme may be used in other layers of this standard.

The FL and RL superframe structures for DuplexingMode = TDD44 and for DuplexingMode = TDD63 are illustrated in Figure 35 and Figure 36, respectively.

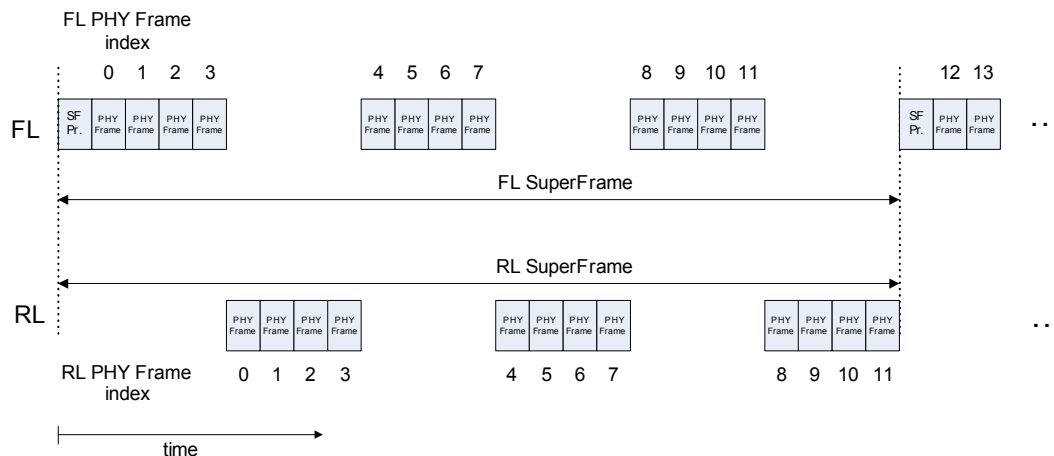


Figure 35—TDD superframe timing for 4:4 partitioning

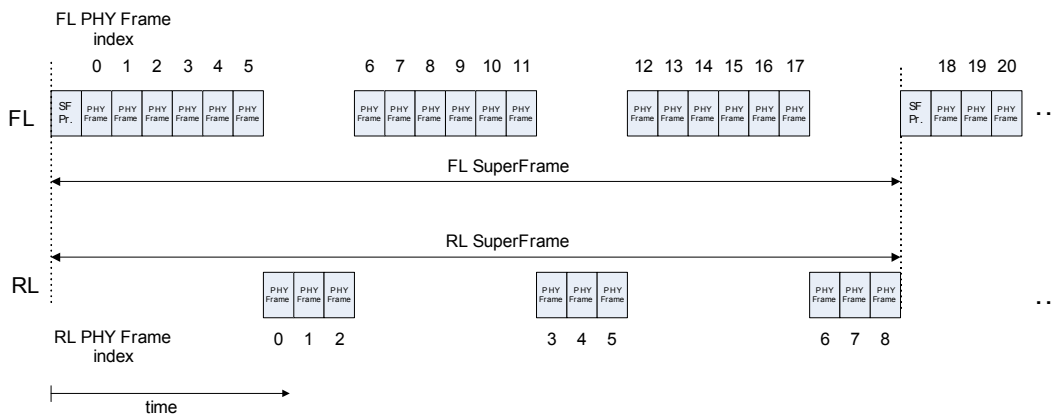


Figure 36—TDD superframe timing for 6:3 partitioning

8.1.5 FDD HARQ interlace structure for the Basic Physical Layer

Both forward and reverse links support HARQ transmission. A HARQ interlacing structure consists of the timing relationship between different transmissions of the same Lower MAC packet and of the acknowledgements for these transmissions.

Multiple HARQ interlacing structures are possible for both the forward and the reverse links. The HARQ interlacing structure can be different for different Lower MAC packets, and is determined by the assignment received on the F-SCCH. This subclause describes the relationship between the timings of the assignments, data transmission, ACK transmission, data retransmissions, etc, for each of the possible interlacing structures.

8.1.5.1 FL interlacing structures

This subclause describes the different interlacing structures that are possible for packets that are transmitted on the FL.

8.1.5.1.1 Eight interlace structure without extended transmissions

If DuplexingMode = FDD, support for this interlacing structure shall be mandatory at the access terminal.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after eight FL PHY frames. Data transmission corresponding to a Forward link assignment that arrives in FL PHY frame k begins in FL PHY frame k . A FL transmission of a Lower MAC packet on FL PHY frame k is acknowledged on RL PHY frame $k + 5$. HARQ transmissions associated with the Lower MAC packet that starts in PHY frame k occur in PHY frames $k + 8n$ where n is the transmission index, $n = 0, 1, \dots$.

Figure 37 shows examples of the timing relationship between FL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each FL PHY frame (i.e., a packet may begin in FL PHY frame 2 and will be acknowledged in RL PHY frame 7 and it will be retransmitted in FL PHY frame 10).

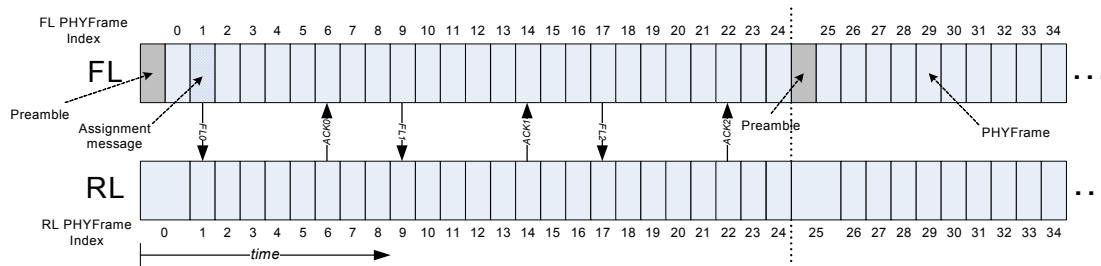


Figure 37—FL 8-interlace structure without extended transmissions

8.1.5.1.2 Six interlace structure without extended transmissions

If DuplexingMode = FDD, support for this interlacing structure shall be optional at the access terminal.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after six FL PHY frames. Data transmission corresponding to a Forward link assignment that arrives in FL PHY frame k begins in FL PHY frame k . A FL transmission of a Lower MAC packet on FL PHY frame k is acknowledged on RL PHY frame $k + 3$. HARQ transmissions associated with the Lower MAC packet that starts in PHY frame k occur in PHY frames $k + 6n$ where n is the transmission index, $n = 0, 1, \dots$.

Figure 38 shows examples of the timing relationship between FL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each FL PHY frame (i.e., a packet may begin in FL PHY frame 2 and will be acknowledged in RL PHY frame 5 and it will be retransmitted in FL PHY frame 8).

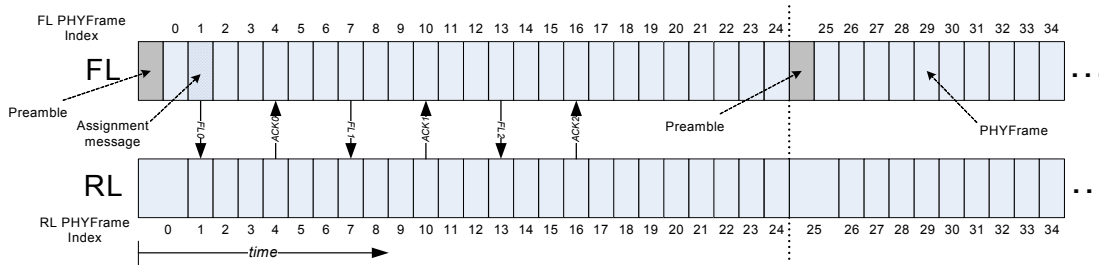


Figure 38—FDD FL 6-interlace structure without extended transmissions

8.1.5.1.3 Eight interlace structure with extended transmissions

If DuplexingMode = FDD, support for this interlacing structure shall be optional at the access terminal.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after eight FL PHY frames. Each HARQ transmission of a data packet spans three FL PHY frames for this interlacing structure. Data transmission corresponding to a Forward link assignment that arrives in FL PHY frame k begins in FL PHY frame k and the first transmission spans FL PHY frames k , $k + 1$, and $k + 2$. A FL transmission of a Lower MAC packet on FL PHY frames k , $k + 1$, and $k + 2$ is acknowledged on RL PHY frame $k + 5$. HARQ transmissions associated with this Lower MAC packet occur in PHY frames $k + 8n$, $k + 8n + 1$ and $k + 8n + 2$ where n is the transmission index, $n = 0, 1, \dots$.

Figure 39 shows examples of the timing relationship between FL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each FL PHY frame (i.e., a packet may begin in FL PHY frame 2; its first HARQ transmission will span FL PHY frames 2, 3, and 4; it will be acknowledged in RL PHY frame 7; and it will be retransmitted in FL PHY frames 10, 11, and 12).

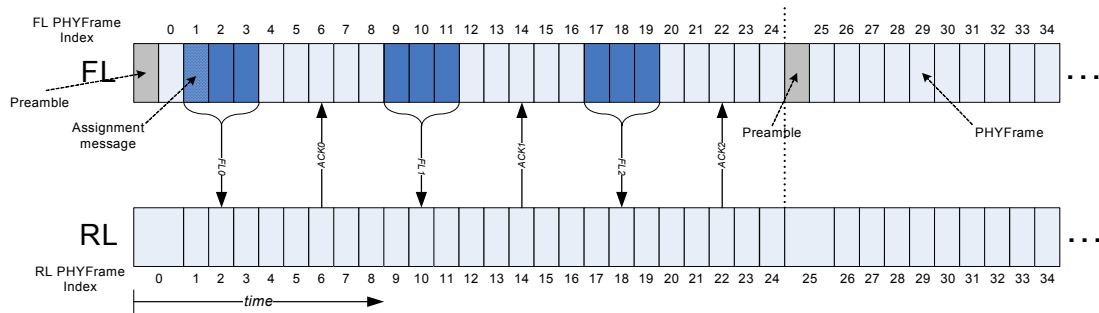


Figure 39—FDD FL 8-interlace structure with extended transmissions

8.1.5.1.4 Eight interlace structure with two frame transmission

If DuplexingMode = FDD, support for this interlacing structure shall be mandatory at the access terminal for access terminals supporting full-duplex operation.

This interlacing structure may be used by the access network only in conjunction with group resource allocation (see 8.6.5.5.1.6 for the definition of *group resource allocation*).

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after seven, eight, or nine FL PHY frames. Each HARQ transmission of a data packet spans up to two consecutive FL PHY frames in this case. A FL transmission of a Lower MAC packet starting in FL PHY frame k spans one or both of PHY frames k and $k + 1$, and is acknowledged on RL PHY frame $k + 5$ or RL PHY frame $k + 6$. The n^{th} HARQ transmission associated with the Lower MAC packet that starts in PHY frame k or PHY frame $k + 1$ spans one or both of PHY frames $k + 8n$ and $k + 8n + 1$ where n is the transmission index, $n = 0, 1, \dots$. The resources assigned to the access terminal through the group resource assignment (GRA) bitmap (see 8.5.5.4.1.1.3.5) determine whether the access network transmits on the first or the second or both PHY frames in the two PHY frame pair. If the access terminal's allocation for a packet contains base-nodes only in PHY frame k , the access terminal shall transmit its acknowledgement information for the packet in PHY frame $k + 5$. If the access terminal's allocation for a packet contains base-nodes in PHY frame $k + 1$ or in both PHY frames, the access terminal shall transmit its acknowledgement information in PHY frame $k + 6$.

Figure 40 shows examples of the timing relationship between FL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each FL PHY frame (i.e., a packet may begin in FL PHY frame 2, its first HARQ transmission will span FL PHY frames 2 or 3 or both, it will be acknowledged in RL PHY frame 7 or 8, and it will be retransmitted in FL PHY frames 10 or 11 or both).

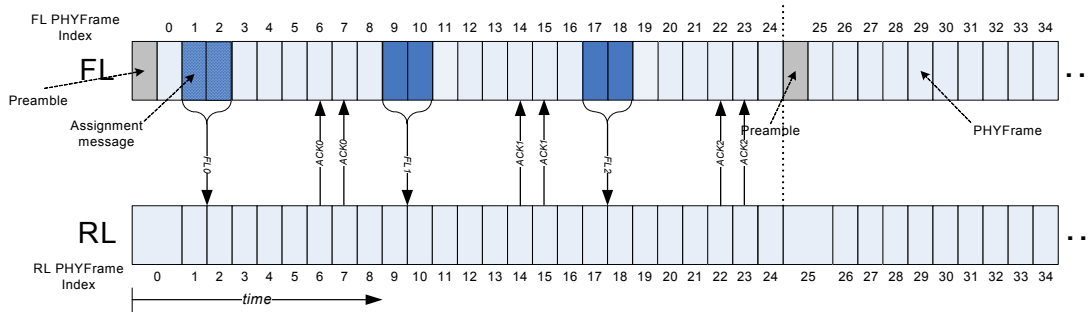


Figure 40—FDD FL 8-interlace structure with two PHY frame transmission

8.1.5.1.5 Six interlace structure with two frame transmission

If DuplexingMode = FDD, support for this interlacing structure shall be optional at the access terminal. If 6-interlace structure is supported by the access terminal, 6-interlace structure with two frame transmission shall also be supported.

This interlacing structure may be used by the access network only in conjunction with group resource transmission. (See 8.5.5.4.1.1.3.5 for the definition of *group resource transmission*.)

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after five, six, or seven FL PHY frames. Each HARQ transmission of a data packet spans up to two consecutive FL PHY frames in this case. A FL transmission of a Lower MAC packet starting in FL PHY frame k spans one or both of PHY frames k and $k + 1$, and is acknowledged on RL PHY frame $k + 3$ or RL PHY frame $k + 4$. The n^{th} HARQ transmission associated with the Lower MAC packet that starts in PHY frame k or PHY frame $k + 1$ spans one or both of PHY frames $k + 6n$ and $k + 6n + 1$ where n is the transmission index, $n = 0, 1, \dots$. The resources assigned to the access terminal through the GRA bitmap (see 8.5.5.4.1.1.3.5) determine whether the access network transmits on the first or the second or both PHY frames in the two PHY frame pair. If the access terminal's allocation for a packet contains base-nodes only in PHY frame k , the access terminal shall transmit its acknowledgement information for the packet in PHY frame $k + 3$. If the access terminal's allocation for a packet contains base-nodes in PHY frame $k + 1$ or in both PHY frames, the access terminal shall transmit its acknowledgement information in PHY frame $k + 4$.

Figure 41 shows examples of the timing relationship between FL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each FL PHY frame (i.e., a packet may begin in FL PHY frame 2, its first HARQ transmission will span FL PHY frames 2 or 3 or both, it will be acknowledged in RL PHY frame 5 or 6, and it will be retransmitted in FL PHY frames 8 or 9 or both).

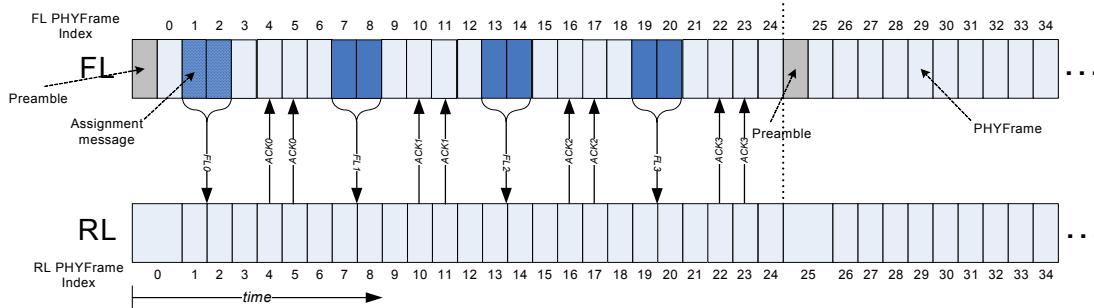


Figure 41—FDD FL 6-interlace structure with two PHY frame transmission

8.1.5.2 RL interlacing structures

This subclause describes the different interlacing structures that are possible for packets that are transmitted on the RL.

8.1.5.2.1 Eight interlace structure without extended transmissions

If DuplexingMode = FDD, support for this interlacing structure shall be mandatory at the access terminal.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after eight RL PHY frames. Data transmission corresponding to a Reverse link assignment that arrives in FL PHY frame k begins in RL PHY frame $k + 3$. A RL transmission of a Lower MAC packet on RL PHY frame $k + 3$ is acknowledged on FL PHY frame $k + 8$. HARQ transmissions associated with the Lower MAC packet that starts in PHY frame $k + 3$ occur in PHY frames $k + 3 + 8n$ where n is the transmission index, $n = 0, 1, \dots$.

Figure 42 shows examples of the timing relationship between RL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each RL PHY frame (i.e., a packet may begin in RL PHY frame 5 and will be acknowledged in FL PHY frame 10 and it will be retransmitted in RL PHY frame 13).

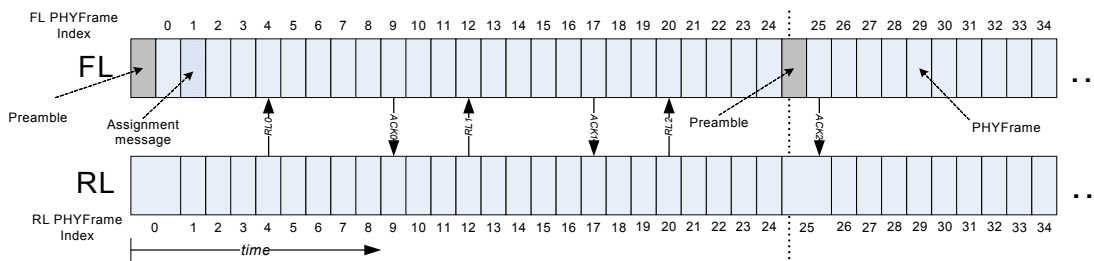


Figure 42—FDD RL 8-interlace structure without extended transmissions

8.1.5.2.2 Eight interlace structure with extended transmissions

If DuplexingMode = FDD, support for this interlacing structure shall be mandatory for access terminals supporting full-duplex operation.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after eight RL PHY frames. Each HARQ transmission of a data packet spans three RL PHY frames for this interlacing structure. Data transmission corresponding to a Forward link assignment that arrives in FL PHY frame k begins in RL PHY frame $k + 3$ and the first transmission spans RL PHY frames $k + 3$, $k + 4$, and $k + 5$. A RL transmission of a Lower MAC packet on RL PHY frames $k + 3$, $k + 4$, and $k + 5$ is acknowledged on FL PHY frame $k + 8$. HARQ transmissions associated with this Lower MAC packet occur in PHY frames $k + 3 + 8n$, $k + 4 + 8n$ and $k + 5 + 8n$ where n is the transmission index, $n = 0, 1, \dots$.

Figure 43 shows examples of the timing relationship between RL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each RL PHY frame (i.e., a packet may begin in RL PHY frame 5; its first HARQ transmission will span RL PHY frames 5, 6, and 7; it will be acknowledged in FL PHY frame 10; and it will be retransmitted in RL PHY frames 13, 14, and 15).

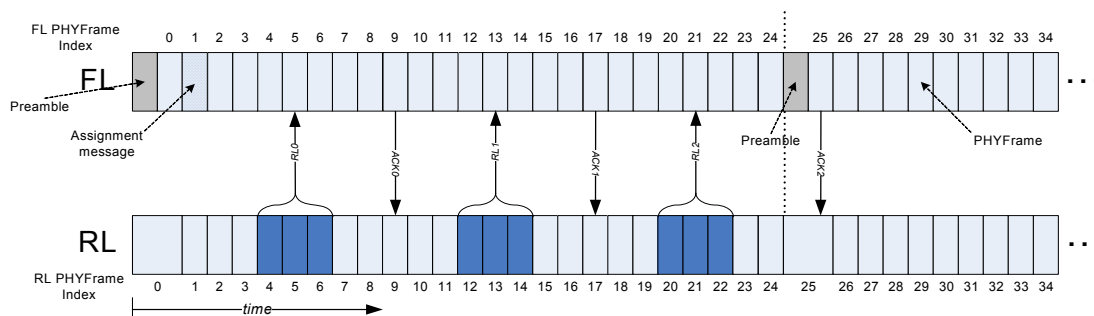


Figure 43—FDD RL Eight interlace structure with extended transmissions

8.1.6 TDD HARQ interlace structure for the Basic Physical Layer

Both forward and reverse links support HARQ transmission. A HARQ interlacing structure consists of the timing relationship between different transmissions of the same Lower MAC packet and of the acknowledgements for these transmissions.

Multiple HARQ interlacing structures are possible for both the forward and the reverse links. The HARQ interlacing structure can be different for different Lower MAC packets, and is determined by the assignment received on the F-SCCH. This subclause describes the relationship between the timings of the assignments, data transmission, ACK transmission, data retransmissions, etc., for each of the possible interlacing structures.

8.1.6.1 TDD HARQ interlace structure for partitioning 1:1

8.1.6.1.1 FL interlacing structures

This subclause describes the different interlacing structures that are possible for packets that are transmitted on the FL for partitioning 1:1.

8.1.6.1.1.1 Four interlace structure

If DuplexingMode = TDD44, support for this interlacing structure shall be mandatory at the access terminal.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after four FL PHY frames. Data transmission corresponding to a Forward link assignment that arrives in FL PHY frame k begins in FL PHY frame k . A FL transmission of a Lower MAC packet on FL PHY frame k is acknowledged on RL PHY frame k . HARQ transmissions associated with the Lower MAC packet that starts in PHY frame k occur in PHY frames $k + 4n$ where n is the transmission index, $n = 0, 1, \dots$.

Figure 44 shows examples of the timing relationship between FL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each FL PHY frame (i.e., a packet may begin in FL PHY frame 2 and will be acknowledged in RL PHY frame 2 and it will be retransmitted in FL PHY frame 6).

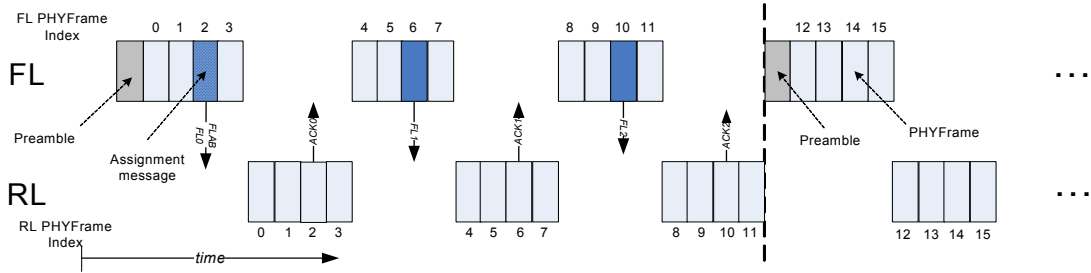


Figure 44—TDD4:4 FL 4-interlace structure

8.1.6.1.1.2 TDD 4:4 FL 4-interlace structure with two PHY frames transmissions

If DuplexingMode = TDD44, support for this interlacing structure shall be mandatory for access terminals.

This interlacing structure may be used by the access network only in conjunction with group resource allocation (see 8.6.5.5.1.1.6 for the definition of Group Resource Allocation).

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after three, four, or five FL PHY frames. Each HARQ transmission of a data packet spans up to two consecutive FL PHY frames that are transmitted in the same burst. The data transmission corresponding to an FL assignment that arrives on FL PHY frame k occurs on one or both of FL PHY frames k and $k + 1$, and is acknowledged on RL PHY frame k or RL PHY frame $k + 1$. Because the FL PHY frames k and $k + 1$ shall be transmitted in the same burst, the resource allocation shall be done such that $k + 1$ is not a multiple of 4.

The n^{th} HARQ transmission associated with the MAC packet that starts in FL PHY frame k or FL PHY frame $k + 1$ spans one or both of FL PHY frames $k + 4n$ and $k + 4n + 1$ where n is the transmission index, $n = 0, 1, \dots$. The resources assigned to the access terminal through the GRA bitmap (see 8.6.5.5.1.1.6) determine whether the access network transmits on the first, the second, or both FL PHY frames in the two PHY frame pair. If the access terminal's allocation for a packet contains base nodes only in FL PHY frame k , the access terminal shall transmit its acknowledgement information for the packet in RL PHY frame k . If the access terminal's allocation for a packet contains base nodes in FL PHY frame $k + 1$ or in both FL PHY frames, the access terminal shall transmit its acknowledgement information in RL PHY frame $k + 1$.

Figure 45 shows examples of the timing relationship between FL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each FL PHY frame (i.e., a packet may begin in FL PHY frame 2, its first HARQ transmission will span FL PHY frames 2 or 3 or both, it will be acknowledged in RL PHY frame 2 or 3, and it will be retransmitted in FL PHY frames 6 or 7 or both).

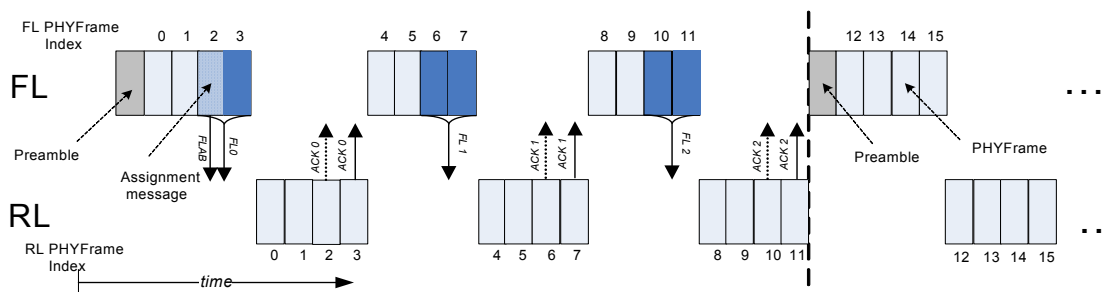


Figure 45—TDD 4:4 FL interlace structure with two PHY frames transmissions

8.1.6.1.2 RL interlacing structures

This subclause describes the different interlacing structures that are possible for packets that are transmitted on the RL for partitioning 1:1.

8.1.6.1.2.1 Four interlace structure without extended transmission

If DuplexingMode = TDD44, support for this interlacing structure shall be mandatory at the access terminal.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after four RL PHY frames. Data transmission corresponding to a Reverse link assignment that arrives in FL PHY frame k begins in RL PHY frame k . A RL transmission of a Lower MAC packet on RL PHY frame k is acknowledged on FL PHY frame $k + 4$. HARQ transmissions associated with the Lower MAC packet that starts in PHY frame k occur in PHY frames $k + 4n$ where n is the transmission index, $n = 0, 1, \dots$

Figure 46 shows examples of the timing relationship between RL packet transmissions and the associated acknowledgement transmissions for this interlacing structure. Note that while the figure shows only one interlace, the interlacing structure applies to packets beginning in each RL PHY frame (i.e., a packet may begin in RL PHY frame 2 and will be acknowledged in FL PHY frame 6 and it will be retransmitted in RL PHY frame 6).

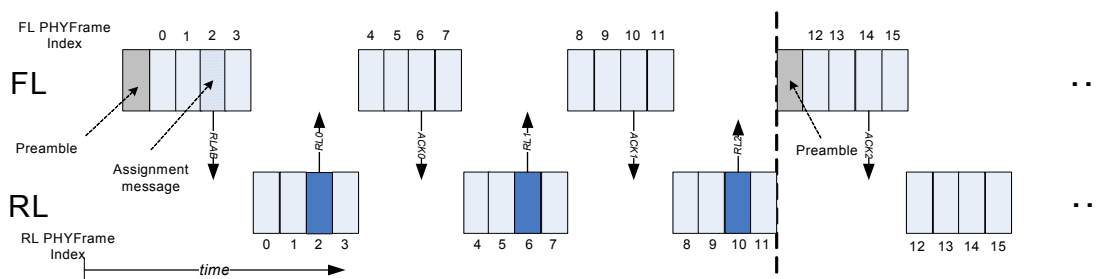


Figure 46—TDD 1:1 RL 4-interlace structures without extended transmission

8.1.6.1.2.2 TDD 4:4 RL 4-interlace structure with extended transmissions

If DuplexingMode = TDD44, support for this interlacing structure shall be optional for access terminals.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after four RL PHY frames. Each HARQ transmission of a data packet spans either the first three RL PHY frames of an RL burst or the last three RL PHY frames of an RL burst. The forward link assignment corresponding to the

data transmission that spans RL PHY frames $4k + p$, $4k + 1 + p$ and $4k + 2 + p$, where $p = 0$ or $p = 1$, is sent in FL PHY frame $4k + 1 + p$.

The data transmission that spans RL PHY frames $4k + p$, $4k + 1 + p$, and $4k + 2 + p$, where $p = 0$ or $p = 1$, is acknowledged in FL PHY frame $4(k + 1) + 1 + p$. HARQ transmissions associated with a MAC packet that starts on RL PHY frames $4k + p$, $4k + 1 + p$, and $4k + 2 + p$, where $p = 0$ or $p = 1$, occur on RL PHY frames $4(k + n) + p$, $4(k + n) + 1 + p$ and $4(k + n) + 2 + p$, $n = 0, 1, \dots$

The data transmission on the RL PHY frames that are not part of an extended transmission follows the rules given in 8.1.6.1.1.1.

Figure 47 shows an example of the timing relationship between RL packet transmissions and the associated acknowledgement transmissions for this interlacing structure.

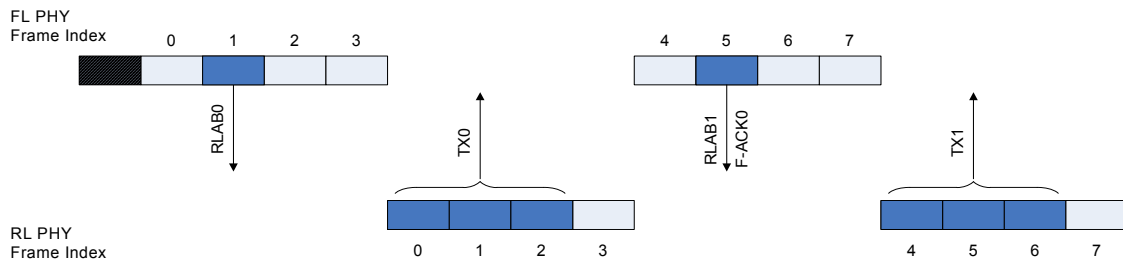


Figure 47—TDD 4:4 RL 4-interlace structure with extended transmissions

8.1.7 TDD 6:3 interlacing structures

8.1.7.1 TDD 6:3 FL interlacing structures

This subclause describes the interlacing structure for packets that are transmitted on the FL for the 6:3 partitioning.

8.1.7.1.1 TDD 6:3 FL 3-interlace structure

If DuplexingMode = TDD63, support for this interlacing structure shall be mandatory for access terminals.

In this interlacing structure, each packet is transmitted over a set of physical channel resources that belongs to a pair of successive FL PHY frames, indexed $2k$ and $2k + 1$. The physical channel resources can belong to only one of the FL PHY frames of the pair or to both FL PHY frames, depending on the assignment. Each HARQ transmission follows the preceding HARQ transmission after a gap of four FL PHY frames. The FL assignments shall be transmitted on even-indexed FL PHY frames. Data transmission corresponding to a Forward link assignment that arrives in FL PHY frame $2k$ begins in FL PHY frame $2k$ and/or $2k + 1$. An FL transmission of a MAC packet on FL PHY frames $2k$ and/or $2k + 1$ is acknowledged on RL PHY frame k . HARQ transmissions associated with the MAC packet that is first transmitted on FL PHY frames $2k$ and/or $2k + 1$ occur in FL PHY frames $2k + 6n$ and/or $2k + 1 + 6n$ where n is the transmission index, $n = 0, 1, \dots$

Figure 48 shows examples of the timing relationship between FL packet transmissions and the associated acknowledgement transmissions for this interlacing structure.

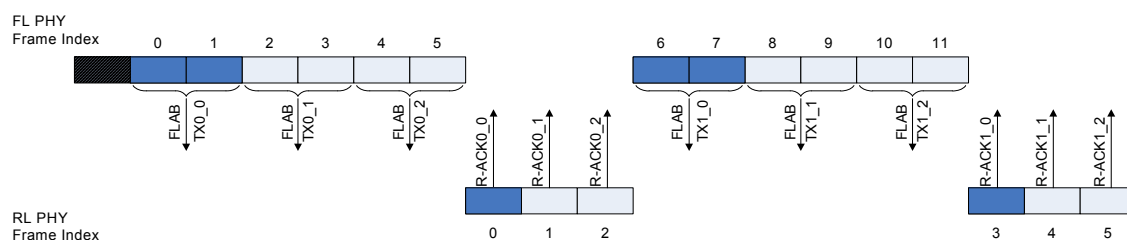


Figure 48—TDD 6:3 FL interlace structure

8.1.7.2 TDD 6:3 RL interlacing structures

This subclause describes the interlacing structure for packets that are transmitted on the RL for the 6:3 partitioning.

8.1.7.2.1 TDD 6:3 RL 3-interlace structure

If DuplexingMode = TDD63, support for this interlacing structure shall be mandatory for access terminals.

In this interlacing structure, each HARQ transmission follows the preceding HARQ transmission after three RL PHY frames. The reverse link assignments are transmitted in odd-indexed FL PHY frames. Data transmission corresponding to a Reverse link assignment that arrives in FL PHY frame $2k + 1$ begins in RL PHY frame k . The corresponding acknowledgment is carried on FL PHY frame $2k + 1 + 6$. HARQ transmissions associated with the MAC packet that starts in RL PHY frame k occur in RL PHY frames $k + 3n$ where n is the transmission index, $n = 0, 1, \dots$

Figure 49 shows examples of the timing relationship between RL packet transmissions and the associated acknowledgement transmissions for this interlacing structure.

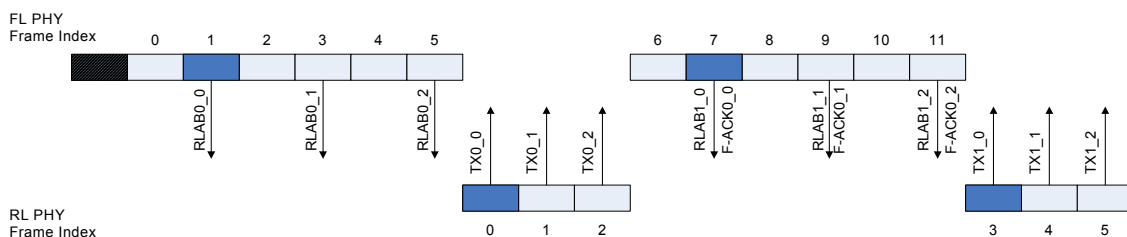


Figure 49—TDD 6:3 RL interlace structure

8.1.7.2.2 TDD 6:3 RL 3-interlace structure with extended transmissions

If DuplexingMode = TDD63, support for this interlacing structure shall be optional for access terminals.

In this interlacing structure, each packet is transmitted over three successive RL PHY frames that belong to the same RL burst and thus are indexed $3k$, $3k + 1$, and $3k + 2$. Each HARQ transmission follows the preceding HARQ transmission in the next group of three RL PHY frames (the next RL burst). The data transmission corresponding to an RL assignment that arrives on the FL PHY frame $6k + 3$ occurs in RL PHY frames $3k$, $3k + 1$, and $3k + 2$. The RL data transmission that occurs in RL PHY frames $3k$, $3k + 1$, and $3k + 2$ is acknowledged on FL PHY frame $6(k + 1) + 3$. HARQ transmissions associated with the MAC

packet that start in RL PHY frames $3k$, $3k + 1$ and $3k + 2$ occur in RL PHY frames $3k + 3n$, $3k + 3n + 1$, $3k + 3n + 2$, where n is the transmission index, $n = 0, 1, \dots$

Figure 50 shows the timing relationship between RL packet transmissions and the associated acknowledgement transmissions for this interlacing structure.

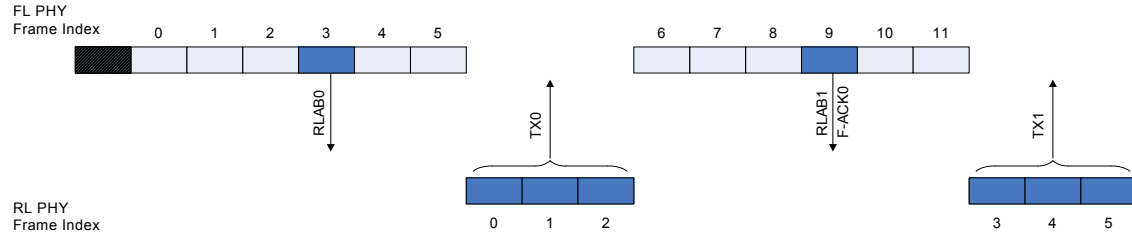


Figure 50—TDD 6:3 RL interlace structure

8.2 Basic Packet Consolidation Protocol

8.2.1 Introduction

8.2.1.1 General overview

The Basic Packet Consolidation Protocol provides packet consolidation on the transmit side and provides packet de-multiplexing on the receive side.

This protocol also negotiates one set of attributes per Stream defined in the Stream Layer. These attributes allow the access terminal to make reverse link scheduling decisions, policing, shaping, and request generation. The attributes in this protocol may only be negotiated by the access network.

The Basic Packet Consolidation Protocol maintains the following two token buckets for each Stream:

- StreamNNSchedulerHintBucketInUse. This bucket is used as a hint for the scheduling of the packets belonging to a Stream on the reverse link.
- StreamNNMaxBucketInUse. This bucket is used for policing and shaping of the traffic flow for a Stream and may be used for scheduling decision on the reverse link.

The access terminal uses these buckets to perform scheduling decisions and request generation along with other attributes defined in this protocol.

This protocol uses the Signaling Protocol to transmit and receive messages.

8.2.2 Primitives and public data

8.2.2.1 Commands

This protocol does not define any commands.

8.2.2.2 Return indications

This protocol does not return any indications.

8.2.2.3 Local common data

This protocol does not define any local common data.

8.2.2.4 Public data

This protocol shall make the following data public:

- Subtype for this protocol.

8.2.2.5 Data encapsulation for the InUse instance of the protocol

The Protocol Data Unit for this protocol is a Consolidated Packet. A Consolidated Packet contains Route Protocol packets destined to or from the same access terminal identifier (ATI).

The Packet Consolidation Protocol encapsulation is shown in Figure 51.

Packets to be transmitted are forwarded to the appropriate Lower MAC Protocol for transmission on the intended Physical Layer channel.

All Route Protocol Packets in received Consolidated Packets are forwarded to the Route Protocol in the Radio Link after removing the Packet Consolidation Header and Packet Consolidation Padding.

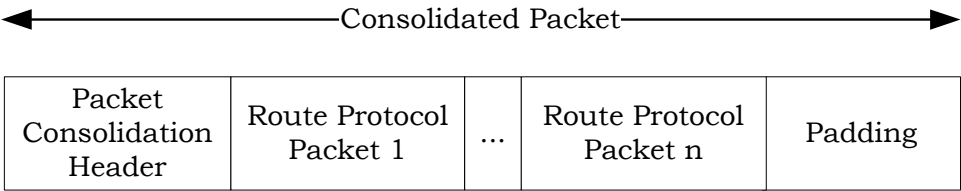


Figure 51—Consolidated Packet structure

8.2.3 Procedures and messages for the InConfiguration instance of the protocol

8.2.3.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

8.2.3.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

8.2.3.3 Message formats

This protocol does not define any messages.

8.2.4 Procedures and messages for the InUse instance of the protocol

8.2.4.1 Initialization and commit procedures

8.2.4.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and the access network shall perform the following in the order specified:

- a) Perform the procedures specified in 15.1.2.
- b) For all Streams, *NN*, the following local variables shall be initialized as follows:
 - $\text{Stream}/\text{NNSchedulerHintBucketInUse} = \text{Stream}/\text{NNSchedulerHintBucket}.\text{BucketSize}/2$
 - $\text{Stream}/\text{NNMaxBucketInUse} = \text{Stream}/\text{NNMaxBucket}.\text{BucketSize}/2$

8.2.4.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

8.2.4.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

8.2.4.2 Other procedures

This protocol receives the following information with every Route Protocol packet to be transmitted:

- Destination channel: Forward Unicast Data Channel, Forward Broadcast Data Channel, or Reverse Unicast Data Channel.
- Priority Order, transmission deadline, and other parameters to help the scheduler.

8.2.4.2.1 Destination channels

Associated with a Route packet received by this protocol, there shall be a parameter indicating the destination channel on which the packet is to be transmitted.

8.2.4.2.2 Priority Order

Associated with a Route packet received by this protocol, there may be a parameter indicating priority order; this parameter is one input in determining the order in which Route packets are chosen for inclusion into a Consolidated packet.

8.2.4.2.3 Transmission deadline

Associated with a Route packet received by this protocol, there may be a parameter indicating a transmission deadline.

8.2.4.2.4 Transmit procedures

The sender shall create a Consolidated packet as described in 8.2.4.4. The resulting packet length shall not exceed the maximum payload that can be carried on the Physical Layer channel, given the transmission rate that will be used to transmit the packet and the headers added by the lower layers. The relative placement of Route packets within a Consolidated packet shall follow the order in which they are to be processed on the

receive side. The policy to determine which Route packets are included in a Consolidated packet is beyond the scope of this standard. It may depend on the Priority Order, transmission deadline, constraints imposed by other protocols, and other criteria related to providing QoS and fairness across reservations.

The Packet Consolidation Protocol at the sender shall forward the Consolidated packet for transmission to the Lower MAC Protocol corresponding to the Physical Layer channel on which the packet is to be transmitted.

8.2.4.2.5 Receive procedures

The receiver shall remove the Packet Consolidation header and forward the Route packets to the Route Protocol.

8.2.4.2.6 Token Bucket Management

8.2.4.2.6.1 Stream*NN*SchedulerHintBucketInUse

For all the Streams, *NN*, the Stream*NN*SchedulerHintBucketInUse shall be incremented every reverse link PHY frame as follows:

$$\text{Stream}NN\text{SchedulerHintBucketInUse} = \min(\text{BucketSize}, \text{Stream}NN\text{SchedulerHintBucketInUse} + \text{TokenRate})$$

where TokenRate and BucketSize are subfields of the Stream*NN*SchedulerHintBucket complex attribute.

When a packet belonging to a Stream *NN* is transmitted, the Stream*NN*SchedulerHintBucketInUse shall be decremented by the number of bytes sent from this Stream.

8.2.4.2.6.2 Stream*NN*MaxBucketInUse

For all the Streams, *NN*, the Stream*NN*MaxBucketInUse shall be incremented every reverse link PHY frame as follows:

$$\text{Stream}NN\text{MaxBucketInUse} = \min(\text{BucketSize}, \text{Stream}NN\text{MaxBucketInUse} + \text{TokenRate})$$

where TokenRate and BucketSize are subfields of the Stream*NN*MaxBucket complex attribute.

When a packet belonging to a Stream *NN* is transmitted, the Stream*NN*MaxBucketInUse shall be decremented by the number of bytes sent from this *Stream*.

8.2.4.2.7 Forward link packet treatment

If the access terminal is constrained in processing packets on the forward link, it shall give higher priority to processing of data for the Streams with higher Stream*NN*FLPriority attribute.

8.2.4.2.8 Reverse link scheduling

The access terminal performs the scheduling of the packets, belonging to a Stream *NN*, on the reverse link using an implementation specific algorithm utilizing the following information:

- Stream*NN*RLPriority attribute (higher value indicates higher priority)
- TransmitAbortTimerRev, which is a public data of Stream *NN*, and the amount of time the packet is in the queue of the Stream *NN*
- Current Stream*NN*SchedulerHintBucketInUse for this Stream
- Current Stream*NN*MaxBucketInUse for this Stream

8.2.4.2.9 Policing and shaping

The access terminal may drop any packet in the queue of a Stream in excess of the StreamNNMaxBucketInUse of that Stream.²¹

The access terminal shall not send more octets from a Stream than specified by the StreamNNMaxBucketInUse of that Stream.

The access terminal may drop packets that stay in the queue of a Stream NN longer than the time limit specified by the TransmitAbortTimerRev attribute for that Stream.

8.2.4.3 Message formats

No messages are defined for the InUse instance of this protocol.

8.2.4.4 Consolidated packet format

The sender shall construct a Consolidated packet as shown in Table 65 and the following text.

Table 65—Consolidated packet format

Field	Length (bits)
n occurrences ($0 \leq n \leq \text{MaxRoutePackets}$) of the following field:	
PacketInfo	8 or 16
HeaderDelimiter	0 or 8
n occurrences ($0 \leq n \leq \text{MaxRoutePackets}$) of the following field:	
Route packet	$8 \times K_i$, where K_i is the value of the i^{th} PayloadLength field, where $0 < i \leq n$.
Padding	As needed

PacketInfo

The number of occurrences of this field (n) shall be equal to the number of Route packets included in this Consolidated packet. The i^{th} PacketInfo record corresponds to the i^{th} Route packet included in this Consolidated packet. The i^{th} occurrence of this field shall be formatted as shown Table 66 and the following text.

Table 66—PacketInfo field format

Field	Length (bits)
More	1
LengthLength	1
PayloadLength	6 or 14

²¹ Note that it is not always desirable to drop packets when this condition is satisfied. For example, dropping packets for stream NN = 0x02 can cause global connection close.

More	This field in the i^{th} occurrence of PacketInfo shall be set to '1' if $i < n$. Otherwise, this field shall be set to '0'.
LengthLength	This field in the i^{th} occurrence of PacketInfo shall be set to '0' if the PayloadLength field in the i^{th} occurrence of PacketInfo is 6 bits long. Otherwise, this field shall be set to '1'.
PayloadLength	This field in the i^{th} occurrence of PacketInfo shall be set to the length, in units of octets, of the i^{th} Route packet in this Consolidated packet. This field shall not be set to zero.
HeaderDelimiter	This field shall be omitted if $n > 0$. If $n = 0$, then this field shall be included and shall be set to 0x00.
RoutePacket	Route packet from the Route Protocol.
Padding	The sender shall add sufficient padding so that the Consolidated packet fills the Lower MAC payload that is available after adding the MAC header and trailers and set the padding bits to '0'. The receiver shall ignore the padding bits.

8.2.4.5 Interface to other protocols

8.2.4.5.1 Commands

This protocol does not issue any commands.

8.2.4.5.2 Indications

This protocol does not register to receive any indications.

8.2.5 Protocol numeric constants

Protocol numeric constants are listed in Table 67.

Table 67—Protocol numeric constants

Constant	Meaning	Value
N _{PCPType}	Type field for this protocol	Table 427
N _{PCPBasic}	Subtype field for this protocol	0x0000

8.2.6 Configuration attributes

8.2.6.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 68. Default values are shown in ***bold italics***.

Table 68—Configurable attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	MaxRoutePackets	Soft Allowed/Not Static	0x10	A Consolidated packet can carry a maximum of sixteen Route packets.
			0x11 to 0xff	Maximum number of Route packets contained in a Consolidated packet.
			0x00 to 0x0f	Reserved.
0x01NN	StreamNNRLLPriority NN is the hexadecimal Stream ID extended to 8 bits by padding '0's on MSBs, where hexadecimal digits A through F are specified in uppercase letters.	Soft Allowed/Dynamic	0x00–0xFF	Relative priority associated with the stream.
			0xFF For streams 1–2	The default priority for signaling streams. NOTE—Default values are not applicable for reserved stream 0.
			All other values	Reserved
0x02NN	StreamNNFLPriority NN is the hexadecimal Stream ID extended to 8 bits by padding '0's on MSBs, where hexadecimal digits A through F are specified in uppercase letters.	Soft Allowed/Dynamic	0x00–0xFF	Relative priority associated with the forward link stream.
			0xFF For streams 0–2	The default priority for forward link signaling streams.
			All other values	Reserved.

8.2.6.2 Complex attributes

The following complex attributes and default values are defined (see 15.2.4 for attribute record definition). This protocol does not define any complex static attributes.

8.2.6.2.1 StreamNNSchedulerHintBucket

NN is the two-digit hexadecimal Stream ID, extended to 8 bits by padding '0's on MSBs, where hexadecimal digits A through F are specified in uppercase letters. This token bucket is used as a hint for the reverse link scheduler (see Table 69).

This attribute has AttributeID 0x81NN.

Table 69—StreamNNSchedulerHintBucket attribute

Field	Length (bits)	Default for Streams 1–2 ^a	Default for all other Streams
TokenRate	16	0x20	0x1000
BucketSize	24	0x0400	0x100000

^aDefault values are not applicable for reserved stream 0.

TokenRate The token rate for the token bucket in terms of octets (as defined in IETF RFC 2216). The token bucket is updated with these tokens every reverse link PHY frame.

BucketSize The size of the bucket for token bucket in terms of octets.

Commit	Soft allowed	Scope	Dynamic

8.2.6.2.2 StreamNNMaxBucket

NN is the two-digit hexadecimal Stream ID, where hexadecimal digits A through F are specified in uppercase letters. This token bucket is used for shaping of the traffic on the reverse link (see Table 70).

This attribute has AttributeID 0x82NN.

Table 70—StreamNNMaxBucket attribute

Field	Length (bits)	Default for streams 1–2 ^a	Default for all other streams
TokenRate	16	0x40	0x2000
BucketSize	24	0x0800	0x200000

^aDefault values are not applicable for reserved stream 0.

TokenRate The token rate for the token bucket in terms of octets (as defined in IETF RFC 2216). The token bucket is updated with these tokens every reverse link PHY frame.

BucketSize The size of the bucket for token bucket in terms of octets.

Commit	Soft allowed	Scope	Dynamic

8.2.7 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of only the configuration attributes of this protocol.

8.3 Basic Superframe Preamble MAC Protocol

The Basic Superframe Preamble MAC Protocol provides the procedures and messages required for an access network to transmit and for an access terminal to receive the following Physical Layer channels: the other-sector interference channel (F-OSICH), and the primary and secondary broadcast control channels (F-PBCCH and F-SBCCH), and the quick paging channel (F-QPCH).

8.3.1 Overview

This standard assumes that the access network has one instance of this protocol for each sector in the network.

This protocol can be in one of the following two states:

- *Inactive State:* In this state, the protocol waits for an Activate command. This state applies only to the access terminal and occurs when the access terminal has not acquired an access network or is not monitoring the Superframe Preamble.

- *Active State*: In this state, the access network transmits and the access terminal receives the channels transmitted on the Superframe Preamble.

The protocol states and allowed transitions between the states are shown in Figure 52.

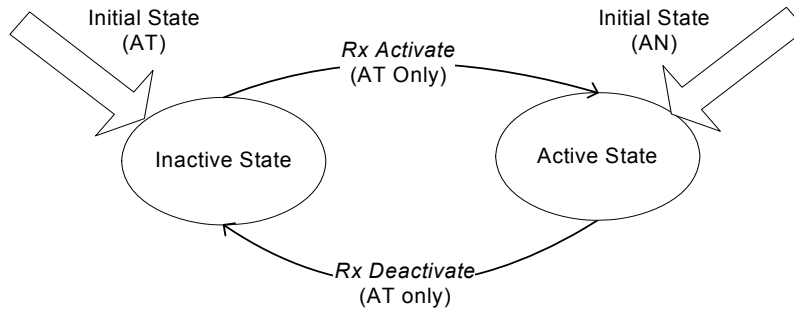


Figure 52—State Diagram of the Basic Superframe Preamble MAC Protocol

This protocol uses the parameters and attributes listed in Table 71.

Table 71—Parameters and attributes for the SFP MAC Protocol

Parameter name	Where defined	Comments
EnableExpandedQPCH	OMP (see 11.6)	QuickChannelInfo
EnableQuickPageOrdering	Configuration attribute	—
QuickPageID	Idle State Protocol	Public data
PagingIDList	Route Control Protocol	Public data

8.3.2 Primitives

8.3.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

8.3.2.2 Return indications

This protocol returns the following indications:

- *PageReceived*
- *QuickPageReceived*

8.3.2.3 Local common data

This protocol does not define any local common data.

8.3.2.4 Public data

This protocol shall make the following data public:

- Subtype for this protocol.

8.3.3 Protocol data unit

The transmission unit of this protocol is defined separately for the four physical channels used by the protocol (F-PBCCH, F-SBCCH, F-QPCH, and F-OSICH), as defined in 8.3.5.6, 8.3.5.7, 8.3.5.8, and 8.3.5.9.

8.3.4 Procedures and messages for the InConfiguration instance of the protocol

8.3.4.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

8.3.4.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

8.3.4.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

8.3.5 Procedures for the InUse instance

8.3.5.1 Initialization and commit procedures

8.3.5.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the InActive State.
- The protocol at the access network shall enter the Active State.

8.3.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

8.3.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

8.3.5.2 Command processing

8.3.5.2.1 Activate

If this protocol receives an *Activate* command in the Inactive State, the access terminal shall transition to the Active State.

If this protocol receives this command in the Active State, the command shall be ignored.

8.3.5.2.2 Deactivate

If this protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

If this protocol receives this command in the Active State, the access terminal shall transition to the Inactive State.

8.3.5.3 Inactive State

This state applies only to the access terminal.

When the protocol is in the Inactive State, the access terminal waits for an Activate command.

8.3.5.4 Active State

In this state, the access network transmits, and the access terminal monitors, the channels associated with the Superframe Preamble MAC.

Procedures for transmission and reception of the F-PBCCH, F-SBCCH, F-QPCH, and F-OSICH Physical Layer channels are specified in 8.3.5.5 through 8.3.5.9.

8.3.5.5 Procedures for transmission and reception of the AcqInfo block

The AcqInfo Block consists of 9 bits and shall be obtained from the Overhead Messages Protocol. Note that this 9-bit value can also be interpreted as an integer between 0 and 511.

At the transmit side, the protocol shall deliver the AcqInfo block to the Physical Layer every superframe for transmission over the TDM Pilot 3 symbol in the superframe preamble (see the Physical Layer specifications for details of the TDM Pilot 3 and the superframe preamble).

At the receive side, the protocol shall deliver the AcqInfo block to the Overhead Messages Protocol.

8.3.5.6 Procedures for transmission and reception of the F-PBCCH Physical Layer Channel

An F-PBCCH transmission has the payload shown in Table 72.

Table 72—F-PBCCH payload

Field	Length (bits)
SystemInfo block	Specified by Overhead Messages Protocol

SystemInfo block The transmitter shall obtain the SystemInfo block from the Overhead Messages Protocol whenever a new F-PBCCH packet is generated.

At the transmit side, the protocol shall deliver the packet to the F-PBCCH for transmission in every superframe. However, a new packet shall be generated only once every $N_{\text{PBCCH-Period}}$ superframes. In Asynchronous mode, a new packet shall be generated in a superframe whose superframe index i satisfies $i \bmod N_{\text{PBCCH-Period}} = 0$. In Synchronous mode, a new packet shall be generated in every superframe where the PilotPhase satisfies $\text{PilotPhase} \bmod N_{\text{PBCCH-Period}} = 0$. The notion of Synchronous and Asynchronous modes is defined by the Physical Layer Protocol.

At the receive side, the protocol shall deliver the received payload to the OMP.

8.3.5.7 Procedures for transmission and reception of the F-SBCCH Physical Layer Channel

An F-SBCCH transmission consists of the payload shown in Table 73.

Table 73—F-SBCCH payload

Field	Length (bits)
QuickChannelInfo block	Specified by Overhead Messages Protocol

QuickChannelInfo block The transmitter shall obtain the QuickChannelInfo block from the Overhead Messages Protocol.

At the transmitter, the protocol shall deliver the packet for transmission on F-SBCCH in every superframe with an odd index.

At the receive side, the protocol shall forward the received payload to the Overhead Messages Protocol.

8.3.5.8 Procedures for transmission and reception of the F-QPCH Physical Layer Channel

An F-QPCH channel transmission consists of one or more instances of the payload shown in Table 74.

Table 74—F-QPCH payload format

Field	Length (bits)
LoadControl	3
QuickPageBlock	35
Reserved	6

LoadControl This field is used by the AC MAC Protocol to determine the class of access terminals that are allowed to make an access attempt in the superframe. This field is used by the AC MAC protocol while processing the AttemptAccess command.

QuickPageBlock The transmitter shall construct the QuickPageBlock in accordance with the reception procedure 8.3.5.8.1.

Reserved The receiver shall ignore this field.

At the transmitter, the protocol shall deliver an ordered set of packets for transmission on F-QPCH in every superframe with an even index. The number of packets to be delivered, denoted by NumQuickPageBlocksPerSF, is given by the following:

- If EnableExpandedQPCH = 0, then NumQuickPageBlocksPerSF = 1.
- If EnableExpandedQPCH = 1, then $\text{NumQuickPageBlocksPerSF} = \lfloor (N_{\text{FFT}} - N_{\text{GUARD}} + 128) / 512 \rfloor$.
Note that EnableExpandedQPCH may not be set to 1 unless $\lfloor (N_{\text{FFT}} - N_{\text{GUARD}} + 128) / 512 \rfloor > 1$.

Here, N_{FFT} and N_{GUARD} are constants of the Physical Layer.

The receiver shall use 8.3.5.8.1 to parse the received QuickPageBlocks.

8.3.5.8.1 Procedure for reception of the QuickPageBlock

The access terminal shall decode the F-QPCH occurring in superframe indices where a QuickPage may be received, as described by the public data of the Idle State Protocol, unless EnableExpandedQPCH is set to '1' and ExpandedQPCHSupported is set to '0'. In the latter case, the access network may page the access terminal without sending a quick page. If both EnableExpandedQPCH and ExpandedQPCHSupported are set to '1', the access terminal shall decode the packet with index equal to the result of applying the hash function (see 15.1.5) using the following parameters:

- *Decorrelate* value of 2 QuickPageID[11:0]
- *N* value of NumQuickPageBlocksPerSF
- *Key* = QuickPageID

where

QuickPageID is public data of the Idle State Protocol
 NumQuickPageBlocksPerSF is defined in 8.3.5.8

The access terminal shall declare the QuickPageBlock to be in error if there is an error in decoding the packet containing this block on the F-QPCH channel (this error information may be used by the Idle State Protocol).

The access terminal shall process each received QuickPageBlock as described below.

The format of the QuickPageBlock shall depend on the first three bits in the block, which are denoted by QuickPageFormat. The actions to be carried out by the access terminal are described in 8.3.5.8.1.1 through 8.3.5.8.1.7 for each of the different values of QuickPageFormat.

8.3.5.8.1.1 QuickPageFormat = '000'

In this case, the protocol shall issue a *PageReceived* indication, accompanied by the received QuickPageFormat, if the PagingID in the QuickPageBlock matches the PagingIDList of the Route Control Protocol. (See Table 75.)

Table 75—QuickPage Format with QuickPageFormat = 0x0

Field	Length (bits)	Value
QuickPageFormat	3	0x0
PagingID	32	ID of paged terminal

8.3.5.8.1.2 QuickPageFormat = '001'

In this case, the protocol shall generate a *QuickPageReceived* indication, accompanied by the received QuickPageFormat, if the least significant bits of QuickPageID that is public data of the Idle State Protocol match one of the QuickPageID_i fields in the QuickPageBlock, where the fields of the QuickPageBlock are as described in Table 76. The number of least significant bits of QuickPageID that are required to match QuickPageID_i is equal to the number of bits in QuickPageID_i. The access terminal may also generate a *PageReceived* indication if an access terminal implementation-dependent algorithm determines that the probability of quick page collision is sufficiently low.

Table 76—QuickPage Format with QuickPageFormat = 0x1

Field	Length (bits)	Value
QuickPageFormat	3	0x1
QuickPageID1	16	QuickPageIDLSBs
QuickPageID2	16	QuickPageIDLSBs

8.3.5.8.1.3 QuickPageFormat = '010'

In this case, the format of the QuickPageBlock is as defined as Table 77. Based on the fields x, y, z, and w, the access terminal shall generate the quantities QuickPageID1, QuickPageID2, and QuickPageID3 as follows:

- If EnableQuickPageOrdering is equal to 0, or if z is equal to w, then QuickPageID1 = x. Otherwise, QuickPageID1 is a 12-bit quantity whose 11 LSBs are given by x, and whose MSB is set to 1 if $z > w$, and whose MSB is set to 0 otherwise.
- QuickPageID2 is given by an 11-bit quantity whose MSB is given by y and whose 10 LSBs are given by z.
- If EnableQuickPageOrdering is equal to 0, then QuickPageID3 = w. Otherwise, QuickPageID3 is given by an 11-bit quantity whose MSB is given by y and whose 10 LSBs are given by w.

The protocol shall generate a *QuickPageReceived* indication, accompanied by the received QuickPageFormat, if the least significant bits of QuickPageID that is public data of the Idle State Protocol match one of the QuickPageID_i fields generated according to the procedure above. The number of least significant bits of QuickPageID that are required to match QuickPageID_i is equal to the number of bits in QuickPageID_i. The protocol may also generate a *QuickPageReceived* indication, accompanied by the received QuickPageFormat, if the least significant bits of QuickPageID that is public data of the Idle State Protocol match one of x, y/z, or w, where y/z denotes an 11-bit quantity with y being the MSB and z being the 10 LSBs.

Table 77—QuickPage Format with QuickPageFormat = 0x2

Field	Length (bits)	Value
QuickPageFormat	3	0x2
x	11	LSBs of QuickPageID1
y	1	MSB of QuickPageID2
z	10	LSBs of QuickPageID2
w	10	LSBs of QuickPageID3

8.3.5.8.1.4 QuickPageFormat = '011'

In this case, the protocol shall generate a *QuickPageReceived* indication, accompanied by the received QuickPageFormat, if the least significant bits of QuickPageID that is public data of the Idle State Protocol match one of the QuickPageID_i fields in the QuickPageBlock, where the fields of the QuickPageBlock are as described in Table 78. The number of least significant bits of QuickPageID that are required to match QuickPageID_i is equal to the number of bits in QuickPageID_i.

Table 78—QuickPage format with QuickPageFormat = 0x3

Field	Length (bits)	Value
QuickPageFormat	3	0x3
QuickPageID1	8	QuickPageIDLSBs
QuickPageID2	8	QuickPageIDLSBs
QuickPageID3	8	QuickPageIDLSBs
QuickPageID4	8	QuickPageIDLSBs

8.3.5.8.1.5 QuickPageFormat = '100' or QuickPageFormat = '101'

In this case, the format of the QuickPageBlock is as defined in Table 79. Define three integers x , y , and z in the range $0 \leq x, y, z < 33$ according to the following procedure:

- Let $b_{29}b_{28}b_{27}...b_0$ be the 30 LSBs of the QuickPageID quantity that is public data of the Idle State Protocol. Define x' , y' , and z' to be integers in the range $0 \leq x', y', z' < 1024$, with their 10-bit binary representations being given by $b_{29}b_{28}...b_{20}$, $b_{19}b_{18}...b_{10}$, and $b_9b_8...b_0$ respectively.
- Define $x = x' \bmod 33$.
- Define $y'' = y' \bmod 32$. Define $y = y''$ if $y'' < x$ and $y = y'' + 1$ if $y'' \geq x$.
- Define $z'' = z' \bmod 31$. Define $\min_{xy} = \min(x, y)$ and $\max_{xy} = \max(x, y)$. Define $z = z''$ if $z'' < \min_{xy}$, $z = z'' + 1$ if $\min_{xy} \leq z'' < \max_{xy} - 1$, and $z = z'' + 2$ if $z'' \geq \max_{xy} - 1$.

The protocol shall generate a *QuickPageReceived* indication, accompanied by the received QuickPageFormat, if all of the bits p_x , p_y , and p_z in the PageIndicatorBitMask field of the QuickPageBlock are set to 1.

Table 79—QuickPage Format with QuickPageFormat = 0x4 or 0x5

Field	Length (bits)	Value
QuickPageFormatMSBs	2	0x2
PageIndicatorBitMask	33	$p_{32}p_{31}...p_0$

8.3.5.8.1.6 QuickPageFormat = '110'

In this case, the format of the QuickPageBlock is as defined in Table 80. Define two integers x and y in the range $0 \leq x, y < 32$ according to the following procedure:

- Let $b_{14}b_{13}b_{12}...b_0$ be the 15 LSBs of the QuickPageID quantity that is public data of the Idle State Protocol.
- Define x in terms of its 5-bit binary representation as $b_{14}b_{13}...b_{10}$.
- Define y' to be an integers in the range $0 \leq y' < 1024$, with its 10-bit binary representations being given by $b_9b_8...b_0$.
- Define $y'' = y' \bmod 31$. Define $y = y''$ if $y'' < x$ and $y = y'' + 1$ if $y'' \geq x$.

The protocol shall generate a *QuickPageReceived* indication, accompanied by the received QuickPageFormat, if both the bits p_x and p_y in the PageIndicatorBitMask field of the QuickPageBlock are set to 1.

Table 80—QuickPage Format with QuickPageFormat = 0x6

Field	Length (bits)	Value
QuickPageFormat	3	0x6
PageIndicatorBitMask	32	$p_{31}p_{30}\dots p_0$

8.3.5.8.1.7 QuickPageFormat = '111'

In this case, the format of the QuickPageBlock is as defined in Table 81. Define an integer x in the range $0 \leq x < 32$ whose binary representation is given by the five LSBs of the QuickPageID quantity that is public data of the Idle State Protocol. The protocol shall generate a *QuickPageReceived* indication, accompanied by the received QuickPageFormat, if the bits p_x in the PageIndicatorBitMask field of the QuickPageBlock is set to 1.

Table 81—QuickPage Format with QuickPageFormat = 0x7

Field	Length (bits)	Value
QuickPageFormat	3	0x7
PageIndicatorBitMask	32	$p_{31}p_{30}\dots p_0$

8.3.5.9 Procedures for transmission and reception of the F-OSICH Physical Layer Channel

The access network shall forward OSI value to the Physical Layer for transmission over the F-OSICH Physical Layer channel. The computation of the OSI value is beyond the scope of this standard.

An F-OSICH channel transmission has the following payload shown in Table 82.

Table 82—F-OSICH Payload format

Field	Length (bits)
OSIValue	2
Reserved	6

OSIValue The value shall indicate the level of interference at the sector. The value '11' is reserved. The OSIValue may also be referred to as an integer between 0 and 2.

Reserved The receiver shall ignore this field.

The access terminal shall monitor the F-OSICH channels of each sector in the OSIMonitorSet (see 8.8.5.5.1.9.3 for details) and provide the received values to the Reverse Traffic Channel MAC Protocol.

8.3.6 Header and trailer formats

The headers and trailers for the F-PBCCCH, F-SBCCCH, F-QPCH, and F-OSICH are described in the subclauses for procedures for transmission and reception for the respective channels, 8.3.5.6, 8.3.5.7, 8.3.5.8, and 8.3.5.9.

8.3.7 Interface to other protocols

8.3.7.1 Commands

This protocol does not issue any commands.

8.3.7.2 Indications

This protocol does not register to receive any indications.

8.3.8 Configuration attributes

8.3.8.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 83.

The access terminal and the access network shall use as default the values in Table 83 listed in ***bold italics***.

Table 83—Configurable attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	EnableQuickPageOrdering	Soft Allowed/Static	<i>0x00</i>	The access terminal does not use the order of the fields in the QuickPageBlock in the reception procedure.
			0x01	The access terminal uses the order of the fields in the QuickPageBlock in the reception procedure.
			All other values	Reserved
0x0001	ExpandedQPCHSupported Capability: The access network shall not suggest values for this attribute.	Soft Allowed/Static	<i>0x00</i>	The access terminal does not support reception of expanded QPCH. In this case, the access terminal may read full pages without reading QPCH, and the access network may send pages to the access terminal without sending a quick page.
			0x01	The access terminal supports reception of expanded QPCH.
			All other values	Reserved

8.3.9 Protocol numeric constants

Numeric constants are listed in Table 84.

Table 84—Numeric constants of the SFP MAC Protocol

Constant	Meaning	Value
$N_{SFPMPType}$	Type field for this protocol	See Clause 12
$N_{SFPMPBasic}$	Subtype field for this protocol	0x00
$N_{PBCCH-Period}$	Periodicity with which a new F-PBCCH packet is generated	16

8.4 Basic Access Channel MAC Protocol

8.4.1 Overview

The Basic Access Channel MAC Protocol (BACMP) provides the procedures and messages required for an access terminal to transmit, and for an access network to receive, the access probe (see Figure 53). An access probe may be used for initial access or handoff within an Active Set. The access network responds to an access probe with an Access Grant over the Forward Link Control Segment (FLCS) MAC Protocol.

This standard assumes that the access network has one instance of this protocol for each sector in the network.

This protocol can be in one of the following two states at the access terminal:

- *Inactive State*: In this state, the protocol waits for an *Activate* command. This state occurs when the access terminal has not acquired an access network.
- *Active State*: In this state, the access terminal may transmit on the Access Channel.

This protocol can be in only the following state at the access network:

- *Active State*: In this state, the access network monitors the Access Channel.

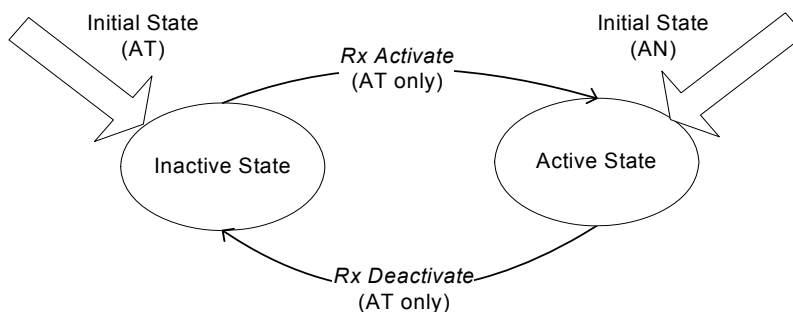


Figure 53—State Diagram for Tree Basic Access Channel MAC Protocol

This protocol uses the parameters and attributes listed in Table 85.

Table 85—Parameters and attributes for the Access Channel MAC Protocol

Parameter name	Where defined	Comments
AccessCycleDuration	OMP	ExtendedChannelInfo
MaxProbesPerSequence	OMP	ExtendedChannelInfo
ProbeRampUpStepSize	OMP	ExtendedChannelInfo
PilotThreshold1	OMP	ExtendedChannelInfo
PilotThreshold2	OMP	ExtendedChannelInfo
OpenLoopAdjust	OMP	ExtendedChannelInfo
AccessRetryPersistence (N _{ACMPClass} values)	OMP	ExtendedChannelInfo
InitialAccessSubsegment	OMP	ExtendedChannelInfo
LoadControl	SFP MAC	F-QPCH
MaxProbeSequences	Configuration attribute	—
PageResponseBackoff	Configuration attribute	—
MaxProbeSequenceBackoff	Configuration attribute	—
SpecialAccessSubsegment	OMP	ExtendedChannelInfo

8.4.2 Primitives

8.4.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*
- *Abort*
- *AttemptAccess*, with the arguments specified as follows:
 - AccessType = {SpecialAccess or InitialAccess}
 - a) If AccessType = InitialAccess, then there is an additional argument InitialAccessType = {ATInit or PageResponse}
 - b) If InitialAccessType = ATInit, then one additional argument shall be included
 - StreamNN
 - c) If InitialAccessType = PageResponse, then two optional arguments may be included
 - AccessPersistenceClass
 - PageResponseAccessSequence

8.4.2.2 Return indications

This protocol returns the following indications:

- *AccessGrantReceived*
- *TransmissionAborted*

- *AccessProbeReceived*
- *AccessFailed*

8.4.2.3 Local common data

This protocol does not define any local common data.

8.4.2.4 Public data

This protocol shall make the following data public:

- Subtype for this protocol
- AccessSequenceID
- AccessType
- PilotID
- ProbePowerDensity
- LastSuccessfulProbeNumber
- LastSuccessfulProbeSequence
- LastAccessedPilotID

8.4.3 Protocol data unit

This protocol does not carry any payload on behalf of other protocols.

8.4.4 Procedures and messages for the InConfiguration instance of the protocol

8.4.4.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

8.4.4.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

8.4.4.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

8.4.5 Procedures for the InUse instance

8.4.5.1 Initialization and commit procedures

8.4.5.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the InActive State.
- The protocol at the access network shall enter the Active State.

8.4.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

8.4.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

8.4.5.2 Command processing

The access network shall ignore all commands.

8.4.5.2.1 Activate

If this protocol receives an *Activate* command in the Inactive State,

- The access terminal shall transition to the Active State.
- The access network shall ignore this command.

If this protocol receives the command in the Active State, the command shall be ignored.

8.4.5.2.2 Deactivate

If this protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

If this protocol at the access terminal receives the command in the Active State, the access terminal shall

- Immediately cease transmitting on the Access Channel if it is in the process of sending a probe.
- Return a *TransmissionAborted* indication if it was in the process of sending an access probe.
- Clear all public data.
- Transition to the Inactive State.

The access network shall ignore this command.

8.4.5.2.3 Abort

If the protocol at the access terminal receives this command, it shall terminate ongoing access procedures and return an *AccessFailed* indication.

8.4.5.2.4 AttemptAccess

If the access terminal receives an *AttemptAccess* command, it shall invoke the procedure for processing an *AttemptAccess* command.

8.4.5.3 Access Channel structure

Figure 54 illustrates the access probe structure and the access probe sequences. In the figure, N_s probe sequences are shown, where each probe sequence has N_p probes. The Access Channel MAC Protocol transmits access probes by instructing the Physical Layer to transmit a probe. With the instruction, the Access Channel MAC Protocol provides the Physical Layer with a power level, a *WalshSequenceID*, an *AccessScramblingID*, and the *PilotID* of the sector to which the access probe is to be transmitted.

Each probe in a sequence is transmitted at an increasing power until any of the following conditions are met:

- The access terminal receives an access grant.
- Transmission is aborted because the protocol received a *Deactivate* or *Abort* command, or for some other reason.
- The maximum number of probes per sequence (MaxProbesPerSequence) has been transmitted. After a maximum number of probes has been transmitted, a new probe sequence may resume from a lower power level.

Prior to the transmission of the first probe of all probe sequences, the access terminal performs a persistence test (see 8.4.5.5.1.3) that is used to control congestion on the Access Channel. This persistence test may return a zero value in some cases, depending on the cause for access.

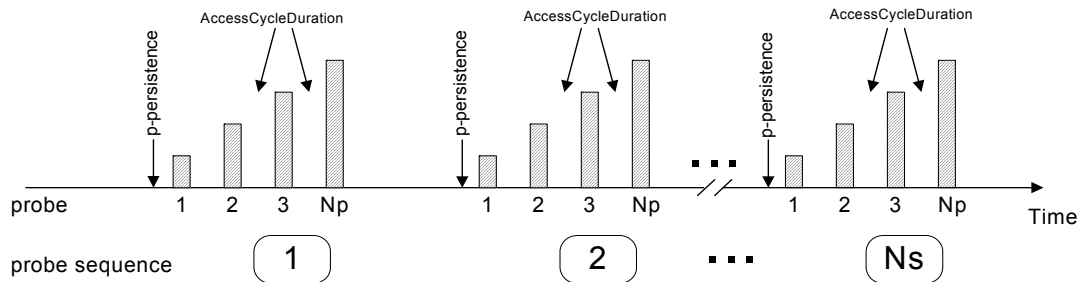


**Figure 54—Access probe sequences
(Ns sequences with Np probes per sequence)**

8.4.5.4 Inactive state

This state applies only to the access terminal.

In this state, the access terminal waits for an *Activate* command.

8.4.5.5 Active state

In this state, the access terminal is allowed to transmit on the Access Channel and the access network is monitoring the Access Channel.

8.4.5.5.1 Access terminal requirements

8.4.5.5.1.1 Procedure for processing the AttemptAccess command

The access terminal shall process only one *AttemptAccess* command at a time. *AttemptAccess* commands shall cause an existing probe transmission to end and a new probe transmission to begin.

If any of the following events occurs while the *AttemptAccess* command is being processed, the access terminal shall invoke the procedure for ending probe transmission 8.4.5.5.1.2.

- The access terminal receives an access grant
(*ForwardLinkControlSegmentMAC.AccessGrantReceived* indication).
- The access grant timer expires.
- The protocol receives an *ActiveSetManagement.IdleHO* or *OverheadMessages.ChannelInfoExpired* indication.

When the procedure for processing the *AttemptAccess* command is invoked, the access terminal shall perform the following initial steps:

- Set ProbeSequenceNumber to 1.
- Set ProbeNumber to 1.
- Set access related parameters as follows:
 - a) If AccessType is SpecialAccess, then
 - AccessPersistenceClass is set to SpecialAccessPersistenceClass attribute.
 - b) If AccessType is InitialAccess and InitialAccessType is ATInit, then
 - AccessPersistenceClass is set to StreamNNAccessPersistenceClass of the StreamNN specified with the command.
 - AccessQoSLevel is set to StreamNNAccessQoSLevel of the StreamNN specified with the command.
 - c) If AccessType is InitialAccess and InitialAccessType is PageResponse, then
 - If a value of AccessPersistenceClass is not specified with the command, then AccessPersistenceClass shall be set to PageResponsePersistenceClass attribute. Otherwise, AccessPersistenceClass is set to the value specified with the command.
 - AccessQoSLevel is set to PageResponseAccessQoSLevel.
- Set TerminalAccessRetryPersistence to the AccessRetryPersistence value indexed by AccessPersistenceClass. For example, if the AccessPersistenceClass is 0, the field shall be set to the first AccessRetryPersistence value in the AccessParametersGroup. The AccessPersistenceClass is set as follows:
 - d) Set PilotID to the Active Set public data field PilotID of the Active Set Management Protocol, or if a PilotID was specified with the *AttemptAccess* command, set PilotID to the specified value.

After performing the initial steps, the access terminal shall perform the following steps to transmit probes:

- a) Determine the WalshSequenceID and the AccessScramblingID by invoking procedure 8.4.5.5.1.4
- b) If ProbeNumber is greater than MaxProbesPerSequence it shall
 - Set ProbeNumber to 1.
 - Increment ProbeSequenceNumber by 1.
- c) If ProbeSequenceNumber is greater than MaxProbeSequences, it shall set an access grant timer for $T_{ACMPANProbeTimeout}$ duration and not perform additional steps.
- d) If ProbeNumber is 1, it shall determine DelayToNextProbe by invoking the procedure for determining probe sequence backoff time 8.4.5.5.1.3. Otherwise, it shall set DelayToNextProbe to the AccessCycleDuration.
- e) Start a counter for DelayToNextProbe RL PHY frames. After the counter is expired, the access terminal shall proceed to the next step. This counter shall be decremented only at the beginning of frames that satisfy the following requirements:
 - 1) The frame contains the CDMA subsegment indexed by InitialAccessSubsegment (when AccessType = 0 or InitialAccess), or contains the CDMA subsegment indexed by SpecialAccessSubsegment (when AccessType = 1 or SpecialAccess). Here, the CDMA subsegment index is the Overall Subsegment Index as defined in 8.7.5.4.7 and InitialAccessSubsegment and SpecialAccessSubsegment are public data of the OMP.
 - 2) The subsegment indexed by InitialAccessSubsegment (if AccessType = 0) or SpecialAccessSubsegment (if AccessType = 1) in this frame does not have any overlap with an RL silence interval. The Physical Layer determines which RL PHY frames and subcarriers are covered by the RL silence interval.

- 3) The LoadControl bits that are transmitted by the SFP MAC as part of the last successfully demodulated F-QPCH instance are set to a value less than or equal to the AccessPersistenceClass. The access terminal shall attempt to decode the F-QPCH while performing access procedures. If the F-QPCH decode attempt is unsuccessful, the access terminal may use the LoadControl bits from the last successfully decoded instance.
- f) Calculate the InitialAccessPowerDensity using the procedures given in 8.4.5.5.1.5.
- g) Transmit a probe using WalshSequenceID, AccessScramblingID, PilotID, and power density calculated as $\text{ProbePowerDensity} = \text{InitialAccessPowerDensity} + \text{ProbeRampUpStepSize} \times (\text{ProbeNumber} - 1)$. The probe shall be sent on the CDMA subsegment indexed by InitialAccessSubsegment, if AccessType = 0, and on the CDMA subsegment indexed by SpecialAccessSubsegment, if AccessType = 1.
- h) Increment ProbeNumber and return to step a).

8.4.5.5.1.2 Procedure for ending probe transmissions

If the access terminal receives an access grant (*ForwardLinkControlSegmentMAC.AccessGrantReceived* indication), it shall

- Terminate the probe transmission procedure.
- Return an *AccessGrantReceived* indication.
- Set the following public data fields:
 - LastSuccessfulProbeNumber to the ProbeNumber of the last probe that has finished transmission.
 - LastSuccessfulProbeSequence to the ProbeSequenceNumber of the last probe that has finished transmission.
 - LastAccessedPilotID to the PilotID from which the access grant was received
 - ProbePowerDensity to the power density at which the last probe was transmitted
- Clear the AccessSequenceID and the AccessType from the public data.
- Terminate this procedure.

If the access grant timer expires, it shall

- Return an *AccessFailed* indication.
- Clear the public data.
- Terminate this procedure.

If the protocol receives an *ActiveSetManagement.IdleHO* indication or an *OverheadMessages.ChannelInfoExpired* indication, it shall

- Return an *AccessFailed* indication.
- Terminate the probe transmission procedure.
- Clear the public data.
- Terminate this procedure.

8.4.5.5.1.3 Procedure for determining probe sequence backoff time

This procedure shall compute the persistence interval, and return an integer value.

If the access attempt was made in response to a page, access was not suspended due to LoadControl or overhead parameter update, and ProbeSequenceNumber is 1, the persistence interval shall be set as follows:

- If the QuickPageFormat field in the QuickPageBlock in the superframe where the access terminal received the page was set to 0x0, the persistence interval shall be set to zero.

- If the QuickPageFormat field in the QuickPageBlock in the superframe where the access terminal received the page was not set to 0x0, the persistence interval shall be set to a random integer drawn uniformly between 0 and $\text{PageResponseBackoff} \times 3$ (both ends inclusive).

Otherwise, if the access attempt was made in response to an *OpenConnection* command from the Air Link Management Protocol at the access terminal, access was not suspended due to LoadControl or overhead parameter update, and the ProbeSequenceNumber is 1, then the persistence interval shall be set to zero.

Otherwise, the persistence interval shall be set as follows:

- If $p < 1$,
 - Generate a geometric random variable n with parameter $p = \text{TerminalAccessRetryPersistence}$, i.e., the random variable takes value n with probability $p(1 - p)^{n-1}$.
 - Set the persistence interval to the minimum of n and $\text{MaxProbeSequenceBackoff}$.
- Otherwise, set the persistence interval to 0.

8.4.5.5.1.4 Procedure for determining WalshSequenceID and AccessScramblingID

The WalshSequenceID is an integer in the range $0 \leq \text{WalshSequenceID} < N_{\text{ACMPWalshSequences}}$, while the AccessScramblingID is an integer between 0 and 15. These quantities have a one-to-one relationship with the AccessSequenceID and the AccessType fields in the public data of this protocol, which will also be described in this subclause.

The procedure for determining WalshSequenceID and AccessScramblingID depends on the AccessType field associated with the received *AttemptAccess* command.

Access probe transmission with AccessType = 0 (InitialConnection) is used for the purpose of obtaining a MACID to start a connection.

Access probe transmission with AccessType = 1 is used for the following purposes and is used only when the access terminal has an assigned MACID on the sector to which the access attempt is made:

- “Hard” handoff between different sectors, either on the same frequency or on different frequencies
- Transitioning to the Connected State from the Semi-Connected State

The values of WalshSequenceID and AccessScramblingID also depend on the values of the quantities PilotLevel and QoSClass, which are defined in the following subclauses.

8.4.5.5.1.4.1 Procedure for determining PilotLevel

To determine the PilotLevel, the access terminal shall use PilotThreshold1 and PilotThreshold2 as described below.

The access terminal shall select its PilotLevel based on the ratio (measured in dB) of the F-PBCCH power from the sector where the access attempt is being made to the total power received in the F-PBCCH time slot (F-PBCCH power plus interference power).

- If this ratio is below PilotThreshold1, the PilotLevel is set to 0.
- If the ratio is not below PilotThreshold1 and is below PilotThreshold2, the PilotLevel is set to 1.
- Otherwise the PilotLevel is set to 2.

The F-PBCCH power referred to previously is the received power of the F-PBCCH, as described in the Physical Layer specification (OFDM symbol with index 0 within the superframe preamble). The F PBCCH time slot refers to the same symbol, i.e., the OFDM symbol with index 0 in the superframe preamble.

8.4.5.5.1.4.2 Determining WalshSequenceID and AccessScramblingID when AccessType = 0 (Initial Access)

In this case, the AccessScramblingID shall be set to 0. The AccessType field in the public data of this protocol shall also be set to 0.

The space of possible WalshSequenceIDs in this case shall be partitioned into $N_{ACMP\text{PilotLevels}} \times N_{ACMP\text{QoSLevels}} + 1$ partitions. Partition 0 shall comprise WalshSequenceIDs in the range $0 \leq \text{WalshSequenceID} < N_{ACMP\text{PilotLevels}} \times N_{ACMP\text{ReservedBaseAccessSequences}}$. For $1 \leq i \leq N_{ACMP\text{PilotLevels}} \times N_{ACMP\text{QoSLevels}}$, the partition i shall comprise WalshSequenceIDs in the range $N_{ACMP\text{PilotLevels}} \times N_{ACMP\text{ReservedBaseAccessSequences}} + (i - 1) \times \text{PartitionSize} \leq \text{WalshSequenceID} < N_{ACMP\text{PilotLevels}} \times N_{ACMP\text{ReservedBaseAccessSequences}} + i \times \text{PartitionSize}$, where PartitionSize is given by $\lfloor (N_{ACMP\text{WalshSequences}} - N_{ACMP\text{PilotLevels}} \times N_{ACMP\text{ReservedBaseAccessSequences}}) / (N_{ACMP\text{PilotLevels}} \times N_{ACMP\text{QoSLevels}}) \rfloor$.

If the PageResponseAccessSequence received along with the AttemptAccess command is not equal to NULL, and the AttemptAccess command was received less than two superframes ago, then the WalshSequenceID shall be set to $\text{PageResponseAccessSequence} \times N_{ACMP\text{PilotLevels}} + \text{PilotLevel}$.

Otherwise, the access terminal shall select a WalshSequenceID with uniform probability over partition number $N_{ACMP\text{PilotLevels}} \times \text{AccessQoSLevel} + \text{PilotLevel} + 1$, where AccessQoSLevel is a field included with the AttemptAccess command.

Here, PilotLevel is determined according to the procedure described in 8.4.5.5.1.4.1 and AccessQoSLevel is a number in the range $0 \leq \text{AccessQoSLevel} < N_{ACMP\text{QoSLevels}}$ and is provided as part of the AttemptAccess command.

The AccessSequenceID in the public data of this protocol shall be set equal to the WalshSequenceID.

8.4.5.5.1.4.3 Determining WalshSequenceID and AccessScramblingID when AccessType = 1

In this case, the AccessType and AccessSequenceID fields in the public data of this protocol shall be set equal to 1 and PilotLevel respectively, where PilotLevel is determined according to the procedure in 8.4.5.5.1.4.1.

The WalshSequenceID and the AccessScramblingID depend on the MACID of the access terminal from the target sector of the access probe.

The WalshSequenceID shall be set to $(\text{MACID} \times N_{ACMP\text{PilotLevels}} + \text{AccessSequenceID}) \bmod N_{ACMP\text{WalshSequences}}$, while the AccessScramblingID shall be set to $\lfloor (\text{MACID} \times N_{ACMP\text{PilotLevels}} + \text{AccessSequenceID}) / N_{ACMP\text{WalshSequences}} + 1 \rfloor$.

8.4.5.5.1.5 Procedure for determining InitialAccessPowerDensity

The parameter InitialAccessPowerDensity shall be determined based on the OpenLoopAdjust parameter and the received power density of the pilot from the sector where the access attempt is being made.

$$\text{InitialAccessPowerDensity} = -\text{MeanRxPowerDensity} + \text{OpenLoopAdjust}$$

where the MeanRxPowerDensity is the Reference Received Power Density as described in the Physical Layer Protocol (Clause 9).

8.4.5.5.2 Access network requirements

If the access network receives an access probe with an associated WalshSequenceID and AccessScramblingID, it shall carry out the following procedure:

- If the AccessScramblingID is equal to 0, it shall set AccessType = 0 and AccessSequenceID = WalshSequenceID in its public data.
- If the AccessScramblingID is not equal to 0, it shall set AccessType = 1, AccessSequenceID = $((\text{AccessScramblingID} - 1) \times N_{\text{ACMPWalshSequences}} + \text{WalshSequenceID}) \bmod N_{\text{ACMPPIlotLevels}}$ and $\text{MACID} = \lfloor ((\text{AccessScramblingID} - 1) \times N_{\text{ACMPWalshSequences}} + \text{WalshSequenceID}) / N_{\text{ACMPPIlotLevels}} \rfloor$ in its public data.
- In either case, the access network shall issue an *AccessProbeReceived* indication, and place the AccessSequenceID in its public data.

If the protocol receives a *ForwardLinkControlSegmentMAC.AccessGrantSent* indication, it shall issue the following commands:

- *ReverseTrafficChannelMAC.Activate*
- *ReverseControlChannelMAC.Activate*

The access network should monitor and control the load on the R-ACH. The access network may control the load by adjusting the LoadControl bits.

8.4.6 Header formats

This protocol does not define any headers.

8.4.7 Message formats

This protocol defines the following messages.

8.4.7.1 AccessProbeNumberRequest

The access network shall send this message to request the probe number of the last successful access probe sent by the access terminal. (See Table 86.)

Table 86—AccessProbeNumberRequest Message

Field	Length (bits)
MessageID	8

MessageID This field shall be set to 0x00.

Channels	FTC
Addressing	Unicast

RLP	Best Effort
AuthTag	Required when key is available

8.4.7.2 AccessProbeNumberResponse

This message shall be sent by the access terminal in response to the AccessProbeNumberRequest message to inform the access network of the probe number that resulted in the reception of the last access grant (see Table 87).

Table 87—AccessProbeNumberResponse message

Field	Length (bits)
MessageID	8
LastAccessedPilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
LastSuccessfulProbeSequence	6
LastSuccessfulProbeNumber	8
Reserved	7

- MessageID

This field shall be set to 0x01.
- LastAccessedPilotID

The access terminal shall set this field to the LastAccessedPilotID public data of the Access Channel MAC.
- ChannelBandIncluded

This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The access terminal shall set this field to 1 if the Combined Active Set includes more than one ChannelBand.
- ChannelBand

This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector where the access terminal performed the previous access. This field shall be omitted otherwise.
- LastSuccessfulProbeSequence

The access terminal shall set this field to the LastSuccessfulProbeSequence public data of the Access Channel MAC.
- LastSuccessfulProbeNumber

The access terminal shall set this field to the LastSuccessfulProbeNumber public data of the Access Channel MAC.
- Reserved

The receiver shall ignore this field.

Channels	RTC	RLP	Best Effort
Addressing	Unicast	AuthTag	Required when key is available

8.4.8 Interface to other protocols

8.4.8.1 Commands

This protocol issues the following commands (access network only):

- *ReverseTrafficChannelMAC.Activate*
- *ReverseControlChannelMAC.Activate*

8.4.8.2 Indications

This protocol registers to receive the following indications:

- *ForwardLinkControlSegmentMAC.AccessGrantSent*
- *ForwardLinkControlSegmentMAC.AccessGrantReceived*
- *ActiveSetManagement.IdleHO*

8.4.9 Configuration attributes

8.4.9.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 88. The access terminal and the access network shall use as default the values in Table 88 listed in ***bold italics***.

Table 88—Simple configuration attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	MaxProbeSequences	Soft Allowed/Static	<i>0x03</i>	The maximum number of probe sequences that can be transmitted as part of one access attempt.
			0x01, 0x02, 0x04, to 0x0f	Other allowed values for this parameter.
			All other values	Reserved.
0x0001	PageResponseBackoff	Soft Allowed/Static	<i>0x01</i>	Expedited response to pages not enabled.
			0x00,	Expedited response to pages enabled.
			All other values	Reserved.
0x0002	MaxProbeSequenceBackoff	Soft Allowed/Static	<i>0x08–0x20</i>	Maximum backoff between probe sequences.
			All other values	Reserved.
0x0003	SpecialAccessPersistenceClass	Soft Allowed/Static	<i>0x00–0x07</i>	Access persistence value to be used for special access.
			All other values	Reserved.
0x0004	PageResponsePersistenceClass	Soft Allowed/Static	<i>0x00–0x07</i>	Access persistence value to be used for initial access in response to a page.
			All other values	Reserved

Table 88—Simple configuration attributes (continued)

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0005	PageResponseAccessQoSLevel	Soft Allowed/Static	0x00–0x01	QoS class to be used by the access terminal in response to a page when no value provided with <i>AttemptAccess</i> command.
			All other values	Reserved
0x01NN	StreamNNAccessQoSLevel <i>NN</i> is the hexadecimal Stream ID extended to 8 bits by padding '0's on MSBs, where hexadecimal digits A through F are specified in uppercase letters.	Soft Allowed/Static	0x00–0x01	The access QoS level to use when an access is made due to a packet in this Stream.
			All other values	Reserved
0x02NN	StreamNNAccessPersistenceClass <i>NN</i> is the hexadecimal Stream ID extended to 8 bits by padding '0's on MSBs, where hexadecimal digits A through F are specified in uppercase letters.	Soft Allowed/Static	0x00–0x07	The access persistence class to use when an access is made due to a packet in this Stream.
			All other values	Reserved

8.4.10 Protocol numeric constants

Numeric Constants of the Access Channel MAC Protocol are listed in Table 89.

Table 89—Numeric Constants of the Access Channel MAC Protocol

Constant	Meaning	Value
N _{ACMPType}	Type field for this protocol	See 16.1
N _{SIACMP}	Subtype field for this protocol	0x00
N _{ACMPClass}	Number of different persistence values	8
N _{ACMPWalshSequences}	Number of Walsh sequences available to the Physical layer to transmit the access probe	1024
N _{ACMPPilotLevels}	Number of PilotLevels used to determine the choice of access sequence	3
N _{ACMPQoSLevels}	Number of QoS classes used to determine the choice of access sequence	2
N _{ACMPReservedBaseAccessSequences}	Number of base access sequences that can be assigned explicitly using a page	16
T _{ACMPANProbeTimeout}	Maximum time to send an acknowledgment for a probe at the access network	5 FL PHYFrames

8.4.11 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

8.5 Basic Forward Link Control Segment MAC Protocol

8.5.1 Overview

The Basic Forward Link Control Segment MAC Protocol provides the procedures and messages required for Lower MAC layer signaling. This includes access network transmissions on the F-SCCH, F-PCCH, F-PQICH, F-FOSICH, F-IOTCH, F-RABCH, F-ACKCH, and F-SPCH Physical Layer channels. These channels together are referred to as the Forward Link Control Segment (FLCS).

This standard assumes that the access network has one instance of this protocol for each sector in the network. However, any implementation that behaves identically is compliant.

This protocol can be in one of the following two states:

- *Inactive State*: In this state, the protocol waits for an *Activate* command. This state applies only to the access terminal and occurs when the access terminal has not acquired an access network or is not monitoring the FL Control Segment.
- *Active State*: In this state, the access network transmits and the access terminal receives the FL Control Segment.

The protocol states and allowed transitions between the states are shown in Figure 55.

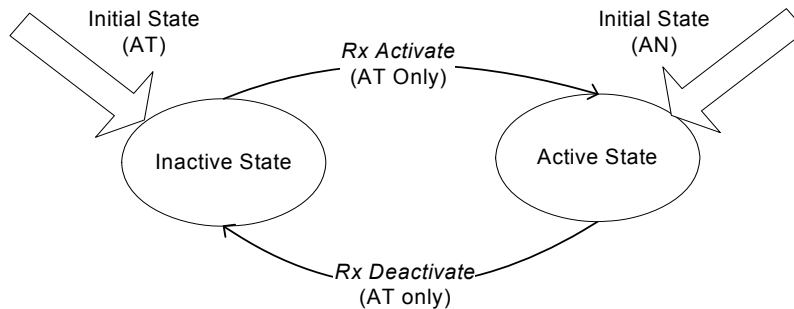


Figure 55—State Diagram for Forward Link Control Segment MAC Protocol

This protocol uses the following parameters and attributes (see Table 90).

**Table 90—Parameters and attributes
for the Forward Link Control Segment MAC Protocol**

Parameter name	Where defined	Comments
AccessSequenceID	Access Channel MAC Protocol	Public data
AccessType	Access Channel MAC Protocol	Public data
BFCHBeamCodeBookIndex	Active Set Management Protocol (ASMP)	Public data
BFCHCodebookType	ASMP	Public data
CDMAFACKIndex	ASMP	Public data
DFLSS	RCC MAC Protocol	Public data
DRLSS	RCC MAC Protocol	Public data
DuplexingMode	OMP	Public data
FACKBandwidthFactor	OMP	ExtendedChannelInfo

**Table 90—Parameters and attributes
for the Forward Link Control Segment MAC Protocol (*continued*)**

Parameter name	Where defined	Comments
FastOSIEnabled	OMP	ExtendedChannelInfo
FLIOTReportInterval	OMP	ExtendedChannelInfo
FLNumSDMADimensions	OMP	QuickChannelInfo
FLPCReportInterval	OMP	ExtendedChannelInfo
FLPQIReportInterval	OMP	ExtendedChannelInfo
FLSS	RCC MAC Protocol	Public data
HalfDuplexMode	Active Sate Management Protocol	Public data
HighSCCHModulationSupported	Configuration attribute	
IoTEnabled	OMP	ExtendedChannelInfo
MACIDRange	OMP	ExtendedChannelInfo
Max16QAMSCCHBlocks	OMP	ExtendedChannelInfo
MaxNumCDMAFACKIndices	OMP	ExtendedChannelInfo
MaxNumSharedLABs	OMP	ExtendedChannelInfo
MaxSCCHDecodedBlocks	Configuration attribute	—
MIMOMode	ASMP	Public data
MinHopportsPerNode	FTC MAC and RTC MAC	Constant
NumEffectiveAntennas	OMP	QuickChannelInfo
NumResourceSets	ASMP	ExtendedChannelInfo
NumResourceSubzones _j , j = 0,...,NumResourceSets – 1	ASMP	ExtendedChannelInfo
PDCABResourceSharingEnabled	OMP	ExtendedChannelInfo
PDCABSupported	Configuration attribute	—
RABEnabled	OMP	ExtendedChannelInfo
ResourceChannelMuxMode	OMP	QuickChannelInfo
RLNumSDMADimensions	OMP	QuickChannelInfo
RLSS	RCC MAC Protocol	Public data
RLSubzoneSize	OMP	ExtendedChannelInfo
SelectedInterlaceAssignment	Connected State Protocol	Public data
SelectedInterlaceMode	Connected State Protocol	Public data
SelectedInterlaceSet	Connected State Protocol	Public data
TotalNumSubcarriers	OMP	SystemInfo
TuneAwayStatus	Air Link Management Protocol	Public data
UserPQIPeriod	ASMP	Public data

where OMP and ASMP refer to the Overhead Messages Protocol and Active Set Management Protocol, respectively.

8.5.2 Primitives

8.5.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

8.5.2.2 Return indications

This protocol returns the following indications:

- *AccessGrantSent* (access network only)
- *AccessGrantReceived* (access terminal only)

8.5.2.3 Local common data

This protocol does not define any local common data.

8.5.2.4 Public data

This protocol shall make the following data public:

- Subtype for this protocol
- AccessGrant block most recently transmitted or received by the protocol
- RPICHScramblingSeed

8.5.3 Protocol data unit

This protocol does not carry payload on behalf of other protocols.

8.5.4 Procedures and messages for the InConfiguration instance of the protocol

8.5.4.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

8.5.4.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

8.5.4.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

8.5.5 Procedures for the InUse Instance

8.5.5.1 Initialization and commit procedures

8.5.5.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the InActive State.
- The protocol at the access network shall enter the Active State.

8.5.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3, in the order specified, when directed to execute the Hard Commit procedures.

8.5.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

8.5.5.2 Command processing

The access network shall ignore all commands.

8.5.5.2.1 Activate

If this protocol receives an *Activate* command in the Inactive State, the access terminal shall transition to the Active State.

If this protocol receives this command in the Active State, the command shall be ignored.

8.5.5.2.2 Deactivate

If this protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

If this protocol receives this command in the Active State, the access terminal shall transition to the Inactive State.

8.5.5.3 Inactive State

This state applies only to the access terminal.

When the protocol is in the Inactive State, the access terminal waits for an *Activate* command.

8.5.5.4 Active State

In this state, the access network transmits, and the access terminal monitors, the Physical Layer channels managed by this MAC protocol.

8.5.5.4.1 Access network requirements

8.5.5.4.1.1 Procedures for the F-SCCH

8.5.5.4.1.1.1 Blocks of the F-SCCH

The F-SCCH can send multiple blocks. Some details about the fields of each block are presented in the descriptions of the blocks that follow the table. A list of fields is provided as follows:

MACID	Sector-specific access terminal identifier. The length of the MACID field is given by $N_{\text{MACID}} = \lceil \log_2 N_{\text{FFT}} \rceil$, where $N_{\text{FFT}} = \text{TotalNumSubcarriers}$.
Persistent	Field used by the assignment blocks to identify if the assignment is persistent or not. A persistent assignment lasts until is explicitly de-assigned, is lost due to packet failure, or expired due to other more recent overlapping assignment. A non-persistent assignment will last for six HARQ transmissions unless successfully decoded before six transmissions or extended by another assignment block.
NodeID	Field used to identify a set of hop-ports. Mapping of NodeID to hop-ports is defined in 8.6.5.6 for the forward link and in 8.8.5.9 for the reverse link. This field has N_{NodeID} bits, where $N_{\text{NodeID}} = \lceil \log_2 (2 \times N_{\text{FFT}} / \text{MinHopPortsPerNode}) \rceil$
SubtreeID	Used to differentiate the SDMA channel subtrees. The number of bits is $N_{\text{STreeIDFL}} = \lceil \log_2 (Q_{\text{SDMAFL}}) \rceil$ or $N_{\text{STreeIDRL}} = \lceil \log_2 (Q_{\text{SDMARL}}) \rceil$ where $Q_{\text{SDMAFL}} = \text{FLNumSDMADimensions}$ $Q_{\text{SDMARL}} = \text{RLNumSDMADimensions}$
SZ	Subzone; field used in SZ-TD-FLAB, SZ-MCW-FLAB1, and SZ-SCW-FLAB to assign one or multiple subzones (as defined by the Physical Layer Protocol) to an access terminal. The structure of the SZ field is described in 8.5.5.4.1.1.3.2.1.
PF	Field used to identify packet formats, and defined in 8.6.5.7 and 8.8.5.10.
RL Pilot	Field used to differentiate two different pilot formats in the RL.
FL Pilot/MIMO	Field used to indicate either the pilot format/rank or precoding information depending on the value of ResourceChannelMuxMode, common/dedicated pilots and the codebook used. The FL Pilot/MIMO field structure is described in 8.5.5.4.1.1.3.2.3 and 8.5.5.4.1.1.3.2.5.
BlockType	This is a 1-bit field; MCW-FLAB2 and RRA-MCW-FLAB1 blocks have the same block header; this field shall be set to 0 if the block is an MCW-FLAB2 and it shall be set to 1 if the block is RRA-MCW-FLAB1.
TD	Transmit Diversity Field; it is present in the assignment blocks that are used to manage the resources for non-precoded F-DCH transmissions. The field indicates the transmit diversity mode used, as described in Table 91. The transmit diversity modes are described in the Physical Layer. This field shall not be set to '001', '010', or '011' if NumEffectiveAntennas is not equal to 2 or 4. This field shall not be set to a value that is inconsistent with the MaxSCWStreams and STTDSupported attribute values for the access terminal to which the assignment block is sent.

Table 91—Interpretation of the TD field

TD field	Transmit diversity mode
'000'	SISO/SIMO

Table 91—Interpretation of the TD field (*continued*)

TD field	Transmit diversity mode
'001'	STTD rate 1
'010'	SM rate 2 when NumEffectiveAntennas = 2 STTD rate 2 when NumEffectiveAntennas = 4
'011'	SM rate 4 when NumEffectiveAntennas = 4 Reserved when NumEffectiveAntennas < 4
'100', '101', '110', '111'	Reserved

HARQ

One-bit field used to distinguish a 6-interlace structure from an 8-interlace structure in FDD. This field shall always be set to 0 in TDD. Different interlacing structures are defined in 8.1.5. The exact interlacing structure is determined in conjunction with the Extended Tx field described below.

Extended TX

One-bit field used to distinguish between extended transmission and non-extended transmission interlacing structures. Different interlacing structures are defined in 8.1.5. The exact interlacing structure is determined in conjunction with the HARQ field described above, according to Table 92. The access network should make sure that the same forward link resources are not assigned to different access terminals. This bit shall always be set to zero in the TDD mode.

Table 92—Interpretation of HARQ and extended Tx fields for FDD

HARQ field	Extended Tx field	Interlacing structure
'0' or Not-present (RLAB)	'0'	Eight interlace structure without extended transmissions.
'0' or Not-present (RLAB)	'1'	Eight interlace structure with extended transmissions.
'1'	'0'	Six interlace structure without extended transmissions.
'1'	'1'	N/A

Table 93—Interpretation of HARQ and extended Tx fields for TDD

HARQ field	Extended Tx field	Interlacing structure
'0' or Not-present (RLAB)	'0'	TDD 4:4 4-interlace structure without extended transmissions or TDD 6:3 3-interlace structure without extended transmissions.
'0' or Not-present (RLAB)	'1'	TDD 4:4 4-interlace structure with extended transmissions or TDD 6:3 3-interlace structure with extended transmissions.
'1'	'0'	N/A

Table 93—Interpretation of HARQ and extended Tx fields for TDD (continued)

HARQ field	Extended Tx field	Interlacing structure
'1'	'1'	N/A

Supplemental	The field will be interpreted together with the NodeID and packet format (PF) fields. If this field is set to 0, the existing assignment is overridden, reassigned, or de-assigned. If the field is set to 1, the existing assignment will be modified incrementally by addition or removal of hop-ports and/or modified through the extension of the HARQ cycle. The exact interpretation of this field is specified in the access terminal assignment (ATA) management procedures of the FTC and RTC MAC protocols (see 8.6.5.5.1.1 and 8.6.5.5.1.3 for details).
RLPC	Reverse Link Power Control Field for RLAB; it indicates the transmit PSD assigned by the access network through a quantized 5-bit number associated with the assigned PF. The exact interpretation of this field is specified in the power control procedures of the RTC MAC protocol.
Timing	This field is only part of the Access Grant Block and indicates the timing offset that the access terminal shall apply to subsequent RL transmissions.
Bitmap	This is a bitmap. Used in PDCAB and GRA.

Table 94—F-SCCH blocks

BLOCK Name	Header (binary)	Fields Length (bits)	Fields	[#bits]
AccessGrant	000	$10 + N_{\text{MACID}}$	MACID Timing	$[N_{\text{MACID}}]$ [10]
RLAB	001	$13 + N_{\text{NodeID}} + N_{\text{STreeIDRL}}$	Persistent NodeID SubtreeID PF RL Pilot Extended Tx Supplemental RLPC	[1] $[N_{\text{NodeID}}]$ $[N_{\text{STreeIDRL}}]$ [4] [1] [1] [1] [5]
TD-FLAB	010	$11 + N_{\text{NodeID}}$	Persistent NodeID PF TD HARQ Extended Tx Supplemental	[1] $[N_{\text{NodeID}}]$ [4] [3] [1] [1] [1]
SZ-TD-FLAB	011	17	SZ PF TD HARQ Extended Tx	[8] [4] [3] [1] [1]

Table 94—F-SCCH blocks (continued)

BLOCK Name	Header (binary)	Fields Length (bits)	Fields	[#bits]
RRA-TD-FLAB	100	$10 + N_{\text{NodeID}}$	NodeID PF TD HARQ Extended Tx Supplemental	$[N_{\text{NodeID}}]$ [4] [3] [1] [1] [1]
SCW-FLAB	101	$13 + N_{\text{NodeID}} + N_{\text{STreeIDFL}}$	Persistent NodeID SubtreeID PF FL Pilot/MIMO HARQ Extended Tx Supplemental	[1] $[N_{\text{NodeID}}]$ $[N_{\text{STreeIDFL}}]$ [4] [5] [1] [1] [1]
MCW-FLAB1	101	$13 + N_{\text{NodeID}} + N_{\text{STreeIDFL}}$	Persistent NodeID SubtreeID PF FL Pilot/MIMO HARQ Extended Tx Supplemental	[1] $[N_{\text{NodeID}}]$ $[N_{\text{STreeIDFL}}]$ [4] [5] [1] [1] [1]
MCW-FLAB2	110	13	BlockType PF1 PF2 PF3	[1] [4] [4] [4]
RRA-SCW-FLAB	110	$12 + N_{\text{NodeID}} + N_{\text{STreeIDFL}}$	NodeID SubtreeID PF FL Pilot/MIMO HARQ Extended Tx Supplemental	$[N_{\text{NodeID}}]$ $[N_{\text{STreeIDFL}}]$ [4] [5] [1] [1] [1]
RRA-MCW-FLAB1	110	$13 + N_{\text{NodeID}} + N_{\text{STreeIDFL}}$	BlockType NodeID SubtreeID PF FL Pilot/MIMO HARQ Extended Tx Supplemental	[1] $[N_{\text{NodeID}}]$ $[N_{\text{STreeIDFL}}]$ [4] [5] [1] [1] [1]
SZ-SCW-FLAB	111	$19 + N_{\text{STreeIDFL}}$	SubtreeID SZ PF FL Pilot/MIMO HARQ Extended Tx	$[N_{\text{STreeIDFL}}]$ [8] [4] [5] [1] [1]

Table 94—F-SCCH blocks (continued)

BLOCK Name	Header (binary)	Fields Length (bits)	Fields [bits]
SZ-MCW-FLAB1	111	$19 + N_{\text{STreeIDFL}}$	SubtreeID [$N_{\text{STreeIDFL}}$] SZ [8] PF [4] FL Pilot/MIMO [5] HARQ [1] Extended Tx [1]
PDCAB	001	MaxNumQPSKLABs	bitmap [MaxNumQPSKLABs]
GRA bitmap	011 or None	$\text{GRA}_{\text{SCCH,SIZE}}$	bitmap [$\text{GRA}_{\text{SCCH,SIZE}}$]

8.5.5.4.1.1.2 Framing of F-SCCH blocks

The framing of blocks in F-SCCH packets follows a specific format as shown in Table 95. An F-SCCH packet consists of a 3-bit header followed by reserved bits, followed by the block. The reserved bits that are used to make all the F-SCCH packets the same size. The Physical Layer Protocol shall add the CRC to each packet.

Table 95—F-SCCH block structure

Field	Length (bits)
Header	3
Reserved	variable
Block	variable

Header A field that identifies the subsequent Block fields. See Table 94 for Header values and the respective blocks and block lengths.

Block See Table 94.

Reserved The access network shall set this field to zero. The access terminal shall ignore the block if this field is not set to zero.

For all blocks except the GRA block, the length of this field shall be such that the total number of bits in the F-SCCH packets is equal to the length of the largest block, excluding the PDCAB and the GRA block, without the padding. Denote the length of each F-SCCH packet thus obtained by L_{SCCH} .

The PDCAB shall be padded with zero LSBs or truncated by removing LSBs to achieve an F-SCCH packet length of L_{SCCH} .

For the GRA block, there is no header field if $\text{GRA}_{\text{SCCH,SIZE}} > L_{\text{SCCH}} - 3$. Otherwise, the header field is '011'. Further, if $\text{GRA}_{\text{SCCH,SIZE}} \leq L_{\text{SCCH}} - 3$, the length of the reserved field shall be such that the overall length (including the header) is L_{SCCH} . If $\text{GRA}_{\text{SCCH,SIZE}} > L_{\text{SCCH}}$, the length of the Reserved field shall be such that the length of the F-SCCH packet is $L_{\text{SCCH}} + 10$ bits. The difference in size with other F-SCCH packets is made up by having a smaller CRC length for this packet in the Physical Layer.

For each FL PHY frame, the access network shall decide the specific blocks to be sent on F-SCCH and shall compute the power density at which each F-SCCH block is transmitted for use by the Physical Layer. In addition to this, the access network shall determine the an AccessSequenceID (in the case of Access Grant blocks sent in response to an initial access) or the MACID (in the case of all other blocks). These variables are used by the Physical Layer protocol to scramble the F-SCCH block. The framed blocks shall be passed to the Physical Layer.

Denote the total number of F-SCCH blocks that are to be sent in the current FL PHY frame by TotalNumSCCHBlocks. In FL PHY frame with index f , the AN shall carry out the following operations:

- a) Set a counter FSCBlockIndex = 0
- b) Determine the F-SCCH block that shall be transmitted and the AccessSequenceID or MACID variable—that is either the MACID of the destination or an AccessSequenceID, as described in the previous paragraphs.
- c) Determine the power density FSCPower that is to be used by the PHY Layer for transmission of this F-SCCH block.
- d) Provide the Physical Layer Protocol with the F-SCCH block, FSCBlockIndex, FSCPower, the AccessSequenceID, or MACID, and the frame index f .
- e) Increment FSCBlockIndex.
- f) If FSCBlockIndex < TotalNumSCCHBlocks, go to step b).

Algorithms for choosing which blocks to send and computing the transmit power density for each block are beyond the scope of this standard.

8.5.5.4.1.1.3 Rules for the F-SCCH blocks

8.5.5.4.1.1.3.1 AccessGrant

This block is sent by the access network upon detection of an access sequence. The block allocates to the access terminal a MACID. The Access Grant block is scrambled with the AccessSequenceID used by the access terminal during transmission of the associated access sequence if the AccessType variable is equal to 0, and is scrambled by the MACID of the access terminal that transmitted the access sequence if the AccessType variable is equal to 1. The AccessType and AccessSequenceID is determined by the AC MAC protocol and can be found in the public data of that protocol. The scrambling procedure is defined in the Physical Layer. The Timing field that is part of the AccessGrant block is used to inform the access terminal of the timing offset to be applied to subsequent RL transmissions, and is interpreted as a signed integer.

8.5.5.4.1.1.3.1.1 Procedures for sending an Access Grant

The access network should send an Access Grant block when it receives an *AccessChannelMAC.AccessProbeReceived* indication.

Two variables are placed in the public data of the AC MAC Protocol—the AccessSequenceID and the AccessType. The AccessType is a one-bit quantity that determines whether the received access probe was sent for initial access or an access probe was sent by a user who already has a MACID.

If the AccessType variable placed in the public data of the AC MAC Protocol is '0', indicating that the access sequence was sent by a user for the purpose of initial access, then the AccessGrant shall contain a MACID that is not in use in the sector.

If the AccessType variable placed in the public data of the AC MAC Protocol is '1', indicating that the access sequence was sent by a user who already has a MACID for the purpose of hand-off or for exiting semi-connected state, then the AccessGrant shall contain the MACID that was used to generate the Physical Layer

sequence corresponding to the received access probe. Refer to the Access Channel MAC Protocol for the procedures for determining AccessSequenceID.

When the Access Grant is sent, the protocol shall generate a *ForwardLinkControlSegmentMAC.AccessGrantSent* indication.

The sector shall not send any F-SCCH blocks to the access terminal with a unicast MACID until three FL PHY frames after the FL PHY frame in which the Access Grant is sent.

8.5.5.4.1.1.3.2 Forward Link Assignment Blocks

This paragraph refers to the blocks TD-FLAB, MCW-FLAB1, MCW-FLAB2, SCW-FLAB, SZ-TD-FLAB, SZ-MCW-FLAB1, SZ-SCW-FLAB, RRA-TD-FLAB, RRA-MCW-FLAB1, and RRA-SCW-FLAB. These assignment blocks inform the access terminals about modifications of resources used for communication on the Forward Link. More exactly, they inform the access terminals that hop-ports specified by the NodeID field have been assigned to, or taken from, them and what packet format should be used on the assigned hop-ports. A Forward Link Assignment Block is scrambled with the MACID of the terminal to which it is sent. The exact scrambling procedure is defined in the Physical Layer. Note that a broadcast MACID (as defined in the Protocol numeric constants of the FLCS MAC Protocol) may be used to transmit the FLAB, in which case the FLAB will be demodulated by all access terminals in the sector.

In the following paragraphs, each different type of assignment block is presented.

8.5.5.4.1.1.3.2.1 Subzone Forward Link Assignment Blocks

This paragraph refers to the following blocks: SZ-TD-FLAB, SZ-MCW-FLAB1, and SZ-SCW-FLAB. These blocks contain the SZ (subzone) field that allows the assignment of multiple subzones to an access terminal. All subzone assignments are non-persistent.

The SZ field has the following eight bits:

- RSIndex—resource set index (2 bits)
- NSZ—number of subzones assigned within the resource set (2 bits)
- SZOffset—subzone offset (4 bits)

The access network shall use the following indexing to set the SZ field.

A subzone is a collection of hop-ports as defined by the Physical Layer Protocol. The Physical Layer Protocol also provides an index for each subzone in each FL PHY frame.

A resource set is a collection of (subzone, interlace) pairs as defined in the Physical Layer Protocol. The resource sets are indexed from 0 to NumResourceSets – 1. The number of subzones contained in resource set i on those FL PHY frames in which the resource set is present is given by the parameter NumResourceSubzones $_i$. Let resource set i be composed of subzones $s_0, s_1, \dots, s_{\text{NumResourceSubzones}(i)-1}$, in the FL PHY frame in which the LAB containing the SZ field is being sent, where the previous indices are arranged in increasing order.

With this indexing scheme, the SZ fields are interpreted in the following way. The subzones referred to by a LAB containing the SZ field shall be the subzones of resource set RSIndex with indices $s_{\text{SZOffset}}, s_{\text{SZOffset}+1}, \dots, s_{\text{SZOffset}+\text{NSZ}-1}$. Note that only the subzones with indices 0 through 3 may be indexed by this LAB.

The SZ-TD-FLAB shall not be transmitted in broadcast mode, i.e., the associated MACID shall not be a broadcast MACID. This restriction is imposed by the fact that this block has the same Header as the GRA

block, which is always transmitted in broadcast mode. The access terminal distinguishes between the two of them based on the MACID.

8.5.5.4.1.1.3.2.2 Residual Resource Assignment (RRA) FLABs

This subclause refers to the following assignment blocks: RRA-TD-FLAB, RRA-MCW-FLAB1, and RRA-SCW-FLAB. All RRA assignments shall be non-persistent.

Through an RRA assignment, an access terminal gets assigned a collection of base nodes (NodeIDs) where certain nodes can be assigned to other access terminals through regular (non-RRA) assignments.

The access network uses the GRA bitmap transmitted on either the F-SCCH or the F-DCH to indicate the F-DCH resources within the RRA assignment that are available in the current PHY frame. The available F-DCH resources are those resources that are unused by regular assignments, including resources not assigned to other access terminals as well as resources persistently assigned to other access terminals but idle in the current frame. The exact rules for determining the hop-ports that are used by a residual resource assignment are described in the FTC MAC Protocol (8.6).

The access network should not send RRA assignments with overlapping resources.

8.5.5.4.1.1.3.2.3 Multiple Codeword MIMO Forward Link Assignment Block part 1

This subclause refers to the following assignment blocks: MCW-FLAB1, SZ-MCW-FLAB1, and RRA-MCW-FLAB1. These blocks contain the FL Pilot/MIMO field. This field indicates the precoding scheme used if the common pilot is used for data demodulation, or the Pilot Format if dedicated pilot is used for data demodulation.

If F-CPICH is used for data demodulation, i.e., for all assignments for the case ResourceChannelMuxMode = 1, and for DRCH assignments for the case ResourceChannelMuxMode = 2, then

- a) If a Knockdown precoder is used, then the FLPilot/MIMO field shall carry only one bit of information that selects between two precoding matrices.
- b) If a Readymade precoder is used, then the FLPilot/MIMO field shall contain no information. The type of precoder is determined by the BFCHBeamCodeBookIndex variable.

If F-DPICH is used for data demodulation, i.e., BRCH in MuxMode2, then the FLPilot/MIMO field shall carry only two bits of information that represent the FLPilot Format.

Since the length of the FLPilot/MIMO field is 5 bits, the LSBs shall be set to 0 when not used to carry information.

The SubtreeID field is used to indicate the subtree on which the first MIMO layer will be transmitted. The subsequent MIMO layers, if they exist, will be placed on the next subtrees, as follows.

Index the MIMO layers from 1 to $N_{\text{EFF\_TX\_ANT}}$, where $N_{\text{EFF\_TX\_ANT}} = \text{NumEffectiveAntennas}$. Then, MIMO layer m will be transmitted on the subtree with index $(\text{SubtreeID} + m - 1) \bmod N_{\text{EFF\_TX\_ANT}}$.

The PF field is used to indicate the packet format for MIMO layer 1. The interpretation of the PF is explained in 8.6.5.7.2.

The access network shall not send a block of this type (either part 1 or part 2, which is described in 8.5.5.4.1.1.3.2.4) to an access terminal that has the MaxMCWLayers attribute set to 0. The access network shall not send a block of this type with a broadcast MACID.

The access network shall not set the number of layers in an assignment block of this type to a value larger than the `MaxMCWLayers` attribute of the terminal to which the assignment block is sent.

8.5.5.4.1.1.3.2.4 Multiple Codeword MIMO Forward Link Assignment Block part 2

This is the MCW-FLAB2. This block is transmitted in conjunction with blocks of type MCW-FLAB1, RRA-MCW-FLAB1, and SZ-MCW-FLAB1. This block shall be transmitted immediately following (i.e., with an index that is one larger than) the block it is paired with. This block informs the access terminal of the packet formats to be used for MIMO layers 2, up through $N_{\text{EFF\_TX\_ANT}}$, via the fields $PF_1 \dots PF_{N_{\text{EFF\_TX\_ANT}}-1}$. The interpretation of each packet format field (PF_i) is described in 8.6.5.7.1.1.

8.5.5.4.1.1.3.2.5 Single Codeword MIMO Forward Link Assignment Blocks

This subclause refers to the following blocks: SCW-FLAB, SZ-SCW-FLAB, and RRA-SCW-FLAB. These blocks contain the same FL Pilot/MIMO field as the MCW-FLAB1 blocks. In addition to the precoding scheme/Pilot Format information, which is represented the same as in the MCW-FLAB1 blocks, this field will indicate the rank of the MIMO transmission, which is the number of MIMO layers that are transmitted on the assignment. See the Physical Layer specification for the definition of MIMO layers.

If F-CPICH is used for data demodulation, i.e., for all assignments for the case `ResourceChannelMuxMode` = 1, and for DRCH assignments for the case `ResourceChannelMuxMode` = 2, then:

- a) If `BFCHCodebookType` = 0 and `BFCHBeamCodebookIndex` = 1, i.e., the Fourier based precoder, the last (LSBs) four bits of FL Pilot/MIMO field shall contain a vector bitmap, each bit equal to '1' indicating a selected column of a 4x4 precoding matrix. The MSB indicates the matrix selection.
- b) If `BFCHCodebookType` = 0 and `BFCHBeamCodebookIndex` = 0, i.e., the Binary Unitary codebook, the last (LSBs) four bits of FL Pilot/MIMO field shall contain a vector bitmap, each bit equal to '1' indicating a selected antenna. If the access terminal does not support space-time transmit diversity (STTD) with antenna selection, i.e., `STTDAntennaSelectionSupported` is set to 0, the MSB shall be reserved and set to 0. In addition, if the access terminal supports STTD with antenna selection, i.e., `STTDAntennaSelectionSupported` is set to 1, and if the number of transmit antennae is less than five and greater than two, the MSB is set to '1', and exactly two bits of the next four bits (the vector bitmap) are set to '1', then STTD mode over the two antennae indicated by the Vector Bitmap shall be used.
- c) If `BFCHCodebookType` = 1, i.e., a Readymade precoder is used, then the first two bits of the FL Pilot/MIMO field shall indicate the rank of the transmission. The type of precoder is determined by the `BFCHBeamCodeBookIndex` variable.

If F-DPICH is used for data demodulation, i.e., for BRCH assignments when `ResourceChanelMuxMode` = 2, then the first two bits of the FL Pilot/MIMO field shall represent the FL Pilot Format and the next two bits shall indicate the rank of the transmission, regardless of the precoder used. In addition to this, if Binary Unitary codebook is used, if the access terminal supports STTD with antenna selection, i.e., `STTDAntennaSelectionSupported` is set to 1, and if the number of transmit antennae is less than five and greater than two, then the LSB shall indicate (set to '1') the presence of the STTD (space-time transmit diversity) mode with antenna selection over two antennas.

The FL Pilot Format is interpreted by the Physical Layer Protocol.

Since the length of the FL Pilot/MIMO field is 5 bits, the LSBs shall be set to 0 when not used to carry information.

The access network shall not set the rank in an assignment block of this type to a value larger than the `MaxSCWStreams` attribute of the terminal to which the assignment block is sent.

The access network shall not set the rank in an assignment block of this type sent with a broadcast MACID to a value larger than 1.

8.5.5.4.1.1.3.3 Reverse Link Assignment Block (RLAB)

The Reverse Link Assignment Block (RLAB) informs the receiving access terminal about a modification of resources used for communication on the Reverse Link. More exactly, it informs the access terminals that hop-ports specified by the NodeID field have been assigned to, or taken from, them and what packet format should be used on the assigned hop-ports. The packet format is indicated by the PF field and is described in 8.8.5.10.

The RL Pilot field indicates the pilot format that shall be used for RL transmission.

The Reverse Link Power Control (RLPC) field is used to indicate the access terminal the transmit Power Spectral Density assigned to it by the access network. The exact interpretation of this field is specified in the power control procedures of the RTC MAC protocol (8.8).

The RLAB shall not be transmitted in broadcast mode, i.e., the MACID shall not be set to any broadcast MACID.

8.5.5.4.1.1.3.4 Packet Data Control Assignment Block (PDCAB)

The Packet Data Control Assignment Block (PDCAB) contains a bit mask of unused F-SCCH resources that can be allocated to the data channel. Each bit equal to '0' indicates a resource that is not used to transmit an F-SCCH block in this FL PHY frame.

The length of this field is MaxNumQPSKLABs , the maximum number of QPSK LABs that can be transmitted in one FL PHY frame, as computed by the Physical Layer Protocol. The LABs that are transmitted in the current FL PHY frame have index pairs (i,j) , where i is in the range 0 to $\text{MaxNumQPSKLABs} - 1$, and j is either 0 or 1 (see 8.5.5.4.1.1.3.6). For each $n = 0, \dots, \text{MaxNumQPSKLABs} - 1$, if a LAB that is transmitted in the current FL PHY frame is assigned index pair $(n,0)$ or $(n,1)$, then the bit with index n (where MSB has index 0) of the bitmap shall be set to 1.

The PDCAB shall always be sent in broadcast mode, i.e., the MACID shall always be set to $\text{MACID}_{\text{BROADCAST}}$.

8.5.5.4.1.1.3.5 Group Resource Allocation (GRA) bitmap

The GRA bitmap indicates which resources on subtree 0 are available in the current frame. Each bit with the value equal to 0 indicates an available resource.

The length of the GRA bitmap, denoted by GRA_{SIZE} , is as determined in 8.6.5.5.1.1.6.4.6. The GRA bitmap is transmitted on the SCCH only if GRA_{SIZE} is less than or equal to 32. In this case, the length is denoted as $\text{GRA}_{\text{SCCH,SIZE}}$.

A value of '0' for the bit with index n indicates that the corresponding set of resources, as defined in the Group Assignment Message, is available for use by a GRA or RRA assignment. For F-SCCH transmissions, the GRA bitmap shall always be sent in broadcast mode, i.e., the MACID shall always be set to $\text{MACID}_{\text{BROADCAST}}$.

8.5.5.4.1.1.3.6 Indexing F-SCCH blocks

The F-SCCH blocks that are transmitted in each FL PHY frame are indexed by a pair (i,j) where i takes values from 0 to $\text{MaxNumQPSKLABs} - 1$, and j takes values 0 and 1. F-SCCH blocks can be transmitted

using either QPSK or 16-QAM modulation. j shall take the value 0 for all F-SCCH blocks transmitted using QPSK modulation. For a given index i , the access network may either transmit a single block with QPSK modulation using index $(i,0)$, or two blocks with 16-QAM modulation and indices $(i,0)$ and $(i,1)$. In other words, if a block index $(i,0)$ is transmitted using QPSK modulation, there shall be no block $(i,1)$.

The access network shall not transmit F-SCCH blocks using 16-QAM modulation with a broadcast MACID. The access network shall also not transmit F-SCCH blocks using 16-QAM modulation with a unicast MACID to access terminals that have the value of the attribute HighSCCHModulationSupported is set to '0'. The access network shall also not transmit F-SCCH blocks using 16-QAM modulation if the F-SCCH block has an index (i,j) with $i \geq \text{Max16QAMSCCHBlocks}$.

The access network shall send F-SCCH blocks to a given access terminal only on the indices that are demodulated by that terminal. The set of F-SCCH block indices that are demodulated by an access terminal is as specified in 8.5.5.4.2.2.

8.5.5.4.1.2 Procedures for F-PCCH

F-PCCH stands for Forward Power Control Channel. For each access terminal that contains the sector in the active set, and whose MACID is less than MACIDRange, the access network shall transmit one power control (PC) bit on F-PCCH every FLPCReportInterval frames. The access network shall determine the value of the PC bit for all MACIDs within MACIDRange, and the power density at which each is to be transmitted, for use by the Physical Layer. The power density should be set to 0 for MACIDs that are not assigned to any access terminal, or for access terminals for which this sector is not the RLSS. The Lower MAC Layer Protocol shall pass these values to the Physical Layer Protocol according to the following procedures.

- a) Let the MACIDs be indexed by m , where $m = 0, \dots, \text{MACIDRange} - 1$.
- b) Define the number of Power Control Bits that are to be transmitted in each FL PHY frame.
- c) $\text{TotalFPCReports} = \lceil \text{MACIDRange} / \text{FLPCReportInterval} \rceil$.
- d) In FL PHY frame f ,
 - 1) Set a MACID index $m = f \bmod \text{FLPCReportInterval}$ and another index $k = 0$.
 - 2) For the MACID m determine the PC bit (PCVal) that is to be sent
 - 3) For the MACID m determine the power density FPCPower at which the PC bit shall be sent
 - 4) Provide the Physical Layer Protocol with the MACID m , the PC bit value PCVal, FPCReportIndex = k , FPCPower and the frame index f .
 - 5) Increment m by FLPCReportInterval. Increment k .
 - 6) If $k < \text{TotalFPCReports}$, go to step 2).

The algorithms for determining the values of the PC bits and the power density at which they are to be transmitted are outside the scope of this standard.

Note that the value of FLPCReportInterval is always even. As a result, PC bits for even MACIDs are always transmitted on FL PHY frames with an even index, while PC bits for odd MACIDs are always transmitted on frames with an odd index.

8.5.5.4.1.3 Procedures for F-PQICH

The access network shall use the following procedures for the Forward Pilot Quality Indication Channel (F-PQICH).

For each access terminal that contains the sector in the active set, and whose MACID is less than MACIDRange, the access network shall transmit one Pilot Quality Indication (PQI) value on F-PQICH

every $\text{FLPQIReportInterval} \times \text{UserPQIPeriod}$ frames. The access network shall determine the PQI values for all MACIDs within MACIDRange, and the power density at which each value is to be transmitted, for use by the Physical Layer. The power density should be set to 0 for MACIDs that are not assigned to any access terminal. The MAC Layer Protocol shall pass these values to the Physical Layer Protocol according to the procedures described as follows.

- a) Let the MACIDs be indexed by m , where $m = 0, \dots, \text{MACIDRange} - 1$.
- b) Define the number of PQI bits that are to be transmitted in each PHY frame
- c) $\text{TotalFPQIReports} = \lceil \text{MACIDRange} / \text{FLPQIReportInterval} \rceil$.
- d) In FL PHY frame f ,
 - 1) Set a MACID index $m = f \bmod \text{FLPQIReportInterval}$ and another index $k = 0$.
 - 2) If $\lfloor f / \text{FLPQIReportInterval} \rfloor \bmod \text{UserPQIPeriod} = f_{\text{PHY-HASH}}(m) \bmod \text{UserPQIPeriod}$, where UserPQIPeriod is public data of the ASMP corresponding to the access terminal with MACID m and $f_{\text{PHY-HASH}}$ is the common PHY hash function defined in the Physical Layer Protocol, then
 - For the MACID m determine the FPQIVal that is to be sent. The FPQIVal is a 4-bit value.
 - For the MACID m determine the power density FPQIPower that shall be used for the transmission of the FPQIVal.
 - 3) Otherwise, set both FPQIVal and FPQIPower to 0.
 - 4) Provide the Physical Layer Protocol with the MACID m , FPQIVal, FPQIReportIndex = k , FPQIPower, and the frame index f .
 - 5) Increment m by FLPQIReportInterval. Increment k .
 - 6) If $k < \text{TotalFPQIReports}$, go to step 2); otherwise exit.

The algorithms for determining the PQI values and the power at which they are to be transmitted are outside the scope of this standard.

Note that the value of FLPQIReportInterval is always even. As a result, PQI values for even MACIDs are always transmitted on FL PHY frames with an even index, while PQI values for odd MACIDs are always transmitted on frames with an odd index.

8.5.5.4.1.4 Procedures for F-FOSICH

If FastOSIEnabled is set to '0', this protocol shall set the value of TotalFFOSIReports to 0 and the access network shall not transmit F-FOSICH.

If FastOSIEnabled is set to '1', then, in addition to the F-OSICH, the access network shall transmit one fast OSI value (1 bit) for each subzone on the F-FOSICH of every RL PHY frame. The F-FOSICH channel may be present on every FL PHY frame if DuplexingMode = FDD or DuplexingMode = TDD44 and on odd-indexed FL PHY frames if DuplexingMode = TDD63, where DuplexingMode is public data of the OMP.

Let s be a subzone index, where $s = 0, \dots, \text{NumRLSubzones} - 1$. The notion of a subzone, as well as the indexing of subzones, is defined in the Physical Layer Protocol. NumRLSubzones denotes the number of subzones in the RL PHY frame and is equal to $N_{\text{FFT}} / \text{RLSubzoneSize}$, where RLSubzoneSize is public data of the OMP. For each s , the access network determines OSI(s), which is a 1-bit value. The MAC Layer Protocol shall pass these values to the Physical Layer Protocol according to the following procedures.

- a) Define the number of fast OSI blocks that are to be transmitted in each PHY frame.
- b) $\text{TotalFFOSIReports} = \lceil \text{NumRLSubzones} / 4 \rceil$.
- c) In FL PHY frame f ,

- 1) Set a counter $k = 0$.
- 2) Concatenate the 4 bits $OSI(k \times 4)$, $OSI(k \times 4 + 1)$, $OSI(k \times 4 + 2)$, $OSI(k \times 4 + 3)$ into a 4-bit block $FFOSIVal$. If some of these bits are not defined (e.g., if the index is out of range) set them to '0'.
- 3) Determine the value $FFOSIPower$ that is the power density that shall be used for the transmission of the $FFOSIVal$.
- 4) Provide the Physical Layer Protocol with the $FFOSIVal$, $FFOSIReportIndex = k$, $FFOSIPower$, and the frame index f .
- 5) Increase k by 1.
- 6) If $k < TotalFFOSIReports$, then go to step 2); otherwise exit.

The computation of the OSI value and the $FFOSIPower$ value is beyond the scope of this standard.

8.5.5.4.1.5 Procedures for F-IOTCH

If $IoTEnabled$ is set to '0', this protocol shall set the value of $TotalFIOTReports$ to 0, and the access network shall not transmit F-IOTCH.

If $IoTEnabled$ is set to '1', then, the access network shall transmit one interference offset value (4 bits) for each subzone on the F-IOTCH every $FLIOTReportInterval$ FL PHY frames. The MAC Layer Protocol shall pass these values to the Physical Layer Protocol according to the following procedures.

Let s be a subzone index, where $s = 0, \dots, NumRLSubzones - 1$. The notion of a subzone, as well as the indexing of subzones, is defined in the Physical Layer Protocol. $NumRLSubzones$ denotes the number of subzones in the RL PHY frame and is equal to $N_{FFT}/RLSubzoneSize$, where $RLSubzoneSize$ is public data of the OMP. If $DuplexingMode = TDD63$, the F-IOTCH can be present only on even-indexed FL PHY frames, while in all other case the F-IOTCH can be present on any FL PHY frame.

Denote the maximum number of IoT reports that are to be transmitted in each PHY frame, by $TotalFIOTReports$.

- a) If $DuplexingMode = FDD$, and if $FDDHalfDuplexTDDPartition = '1'$ and $FLIOTReportInterval > 1$, set $TotalFIOTReports = \lceil 2 \times NumRLSubzones / FLIOTReportInterval \rceil$.
- b) Otherwise, set $TotalFIOTReports = \lceil NumRLSubzones / FLIOTReportInterval \rceil$
- c) In FL PHY frame f ,
 - 1) If $DuplexingMode = FDD$ and $FDDHalfDuplexTDDPartition = '1'$ and $FLIOTReportInterval > 1$, initialize a subzone index s to $\lfloor (f \bmod FLIOTReportInterval) / 2 \rfloor$. Otherwise, initialize s to $f \bmod FLIOTReportInterval$.
 - 2) Initialize another counter k to 0.
 - 3) Determine the value $IOTVal$, which is the IoT value for subzone s .
 - 4) Determine $FIOTPower$, which is the power density that shall be used for the transmission of $IOTVal$.
 - 5) Provide the Physical Layer Protocol with the $IOTVal$, $FIOTReportIndex = k$, $FIOTPower$, and the frame index f .
 - 6) If $DuplexingMode = FDD$ and $FDDHalfDuplexTDDPartition = '1'$ and $FLIOTReportInterval > 1$, increment s by $FLIOTReportInterval / 2$. Otherwise, increment s by $FIOTReportInterval$.
 - 7) Increment k .
 - 8) If $k < TotalFIOTReports$, then go to step 3); otherwise exit.

The computation of the IoT value and the $FIOTPower$ value is beyond the scope of this standard.

8.5.5.4.1.6 Procedures for F-RABCH

If RABEnabled is set to '0', this protocol shall set the value of TotalFRABReports to 0, and the access network shall not transmit F-RABCH.

If RABEnabled is set to '1', the access network shall transmit the RAB value (1 bit) on the F-RABCH every FL PHY frame if DuplexingMode = FDD or DuplexingMode = TDD44 and on even-indexed FL PHY frames if DuplexingMode = TDD63. This bit indicates which flows are permitted on the R-CDCH and is interpreted by the RTC MAC protocol. The MAC Layer Protocol shall pass this value to the Physical Layer Protocol according to the following procedures.

Define TotalFRABReports = 1. In FL PHY frame with index f where F-RABCH is transmitted,

- 1) Determine the value RABVal, which is the RAB bit.
- 2) Determine FRABPower, which is the power density that shall be used for the transmission of RABVal.
- 3) Provide the Physical Layer Protocol with the RABVal, FRABPower, FRABReportIndex = 0, and the frame index f .

The computation of the RABVal and the FRABPower value is beyond the scope of this standard.

8.5.5.4.1.7 Procedures for F-ACKCH

Acknowledgements for RL traffic are sent on F-ACKCH.

The relationship between the RL PHY frame, or a set of RL PHY frames, on which a packet is transmitted, and the FL PHY frame carrying the corresponding ACK for each interlacing structure is shown in 8.1.5. The interlacing structure may change from packet to packet and is determined for each packet by the RTC MAC Protocol.

On every FL PHY frame, base nodes in each subtree of the reverse link channel tree on the corresponding RL interlace shall be assigned an FACKNodeIndex according to the following procedure (note that some base nodes are not assigned an FACKNodeIndex):

- 1) Initialize a subtree counter q to 0. Initialize a base node counter b to the lowest indexed base node in channel subtree q . Initialize an FACKNodeIndex counter k to 0.
- 2) If base node b in subtree q is comprised of any usable hop-ports (as defined in the Physical Layer) and is not assigned to the Reverse Link Control Segment (as described in the Physical Layer Protocol), then
 - The FACKNodeIndex of this base node shall be k .
 - Increment k by 1.
- 3) Increment b by FACKBandwidthFactor. If b is larger than the highest indexed based node in subtree q , then
 - Increment q by 1.
 - Set b to the lowest indexed base node in subtree q .
- 4) Repeat step 2) and step 3) until $q = Q_{\text{SDMA}}$, where Q_{SDMA} is equal to RLNumSDMADimensions, which is public data of the OMP.

Denote the total number of base nodes on RL interlace k that received an FACKNodeIndex with TotalOFDMAFACKNodeIndices $_k$.

In addition, an access terminal may be assigned a CDMAFACKIndex in a given FL PHY frame that correspond to RL interlaces k for acknowledging CDMA traffic, i.e., data sent on the R-CDCH. The FACKNodeIndex assigned to an access terminal with CDMAFACKIndex = i in a given FL PHY frame shall be set to TotalOFDMAFACKNodeIndices $_k$ + i .

Denote by TotalFACKNodeIndices the total maximum number of acknowledgement indices in a given FL PHY frame. This number can be derived according Equation ().

$$\text{TotalFACKNodeIndices} = \max_{k=0,\dots,7} \{ \text{TotalOFDMAFACKNodeIndices}_k + \text{NumCDMAFACKIndices}_k \} \quad (1)$$

where NumCDMAFACKIndices $_k$ = MaxNumCDMAFACKIndices if RL interlace k has a CDMA segment allocation, and NumCDMAFACKIndices $_k$ = 0, otherwise, and MaxNumCDMAFACKIndices is public data of the OMP.

Acknowledgements are indicated via a field called FACKValue. There are four possible FACK values denoted by 0, 1, 2, and 3. The FACKValue 0 is indicated by the Physical Layer by blanking the transmission, i.e., by using zero power.

The 0 value indicates no acknowledgement.

The 1 value indicates an acknowledgement; this value is used at the end of packet transmission to inform the access terminal

- That a Lower MAC packet it transmitted on the RL passed CRC.
- That it can keep the persistent assignment for the subsequent packet.

The 2 value indicates both an acknowledgement and a deassignment; it informs the access terminal

- That a Lower MAC packet it transmitted on the RL passed CRC.
- Not to transmit any more packets using the current assignment.

The 3 value indicates a de-assignment only, i.e., that the access terminal should stop transmitting on the current resource. However, it does not indicate that an ongoing packet has been successfully decoded.²²

Detailed rules for managing the access terminal assignment (ATA) on the reverse link based on the transmitted F-ACKCH values are provided in the RTC MAC protocol.

For each access terminal, an ATA of the interlace containing the RL PHY frame is being acknowledged by sending an acknowledgement block for only one base node of the respective ATA. The base node of the respective ATA that has the lowest FACKNodeIndex shall be acknowledged through the FACKValue corresponding to that index. The rest of the base nodes in the access terminal's ATA shall not be acknowledged, i.e, their FACKValue shall be set to zero.

The access network shall also determine the power used to transmit each of these values, for use by the Physical Layer. Algorithms for determining this power are outside the scope of this standard. In addition, for every ACK indication, this protocol shall pass to the Physical Layer a valid MACID corresponding to the acknowledged access terminal, which is needed for MACID scrambling.

²²It shall be used when the user is transmitting keep-alive sequences instead of actual packets. This extra ACK value is required since keep-alive sequences are unreliable; the use of state 2 (ACK and de-assign) in the case of the false keep-alive sequence detection by the access network would result in access terminal observing a false ACK of a packet in progress.

The values shall be sent to the Physical Layer according to the following procedure:

- 1) Initialize an index FACKNodeIndex to 0.
- 2) If the base node with the FACKNodeIndex is part of an RL assignment and all other base nodes of the same assignment have a greater FACKNodeIndex (i.e., is the lowest indexed base node from this assignment), then determine the FACKValue corresponding to this assignment. Otherwise, set the FACKValue of this base node to 0.
- 3) If FACKValue is not equal to 0,
 - Determine the FACKPower corresponding to FACKNodeIndex.
 - Provide the Physical Layer Protocol with FACKValue, FACKNodeIndex, FACKPower, the frame index, and the MACID of the access terminal that used the RL resources indicated by FACKNodeIndex.
- 4) Increment FACKNodeIndex by 1.
- 5) If FACKNodeIndex < TotalOFDMAFACKNodeIndices, go to step 2).

8.5.5.4.1.8 Procedures for F-SPCH

The start-of-packet channel (F-SPCH) is used by the access network to send start-of-packet (SP) indications to the access terminals. An SP indication is used to retain a Forward Link persistent assignment of an access terminal in the absence of active data transmission, or to signal the start of a new packet on the persistent assignment, or to deassign the persistent assignment.

If DuplexingMode = TDD63, the F-SPCH channel is present only on even-indexed FL PHY frames; otherwise, it is present on all FL PHY frames.

The SP indications can have four possible values denoted by 0, 1, 2, 3. The 0 value is mapped by the PHY Layer into a modulation symbol with zero power.

The 0 value means that no indication is sent.

The 1 value is used to inform the access terminal to attempt decoding a new packet. The FL PHY frame on which a new packet is transmitted is the same as the FL PHY frame carrying the SP indication. This value also has the implicit meaning that the access terminal can keep the persistent assignment for the subsequent packet.

The 2 value indicates a deassignment; it is used to inform the access terminal not to attempt any decoding in the current assignment.

The 3 value shall not be used.

Detailed rules for managing the ATA on the forward link based on the transmitted F-SPCH values are provided in the FTC MAC protocol.

The SP indications shall be transmitted to the Physical Layer according to the following procedure.

- a) Selected base nodes of the forward link channel tree shall be assigned an FSPNodeIndex according to the following procedure:
 - 1) Initialize a base node counter b to the smallest base node index in channel subtree 0. Initialize an FSPNodeIndex counter k to 0.
 - 2) If base node b only comprises of usable hop-ports (as defined in the Physical Layer) and both PDCABResourceSharingEnabled and PDCABSupported are set to '1', or, if base node b only comprises of usable hop-ports (as defined in the Physical Layer), either PDCABResourceSharingEnabled or PDCABSupported is set to '0' and base node b is not

assigned to the Forward Link Control Segment (as described in the Physical Layer Protocol), then

- The FSPNodeIndex of this base node on all subtrees shall be k.
 - Increment k by 1.
- 3) Increment b by RACKBandwidthFactor.
 - 4) If b is smaller than the largest base node index in the subtree 0, go to step 2).
- b) Denote the total number of base nodes that received an FSPNodeIndex with TotalFSPNodeIndices.

The access terminal's FSPNodeIndex shall be determined for on each FL interlace. Note that for DuplexingMode = TDD63, the channel tree addresses the resources of a pair of FL PHY frames, as is described in 8.6.5.6. This FSPNodeIndex shall be determined according to the following rule. Let m denote the SubtreeIndex of the forward link access terminal assignment. Let N denote the number of base nodes within the access terminal assignment that are assigned an FSPNodeIndex, and refer to these base nodes as ackable nodes 0, 1, ..., N - 1, where the indexing starts from the base node with the smallest NodeID and successive indices are assigned to base nodes in increasing order of their NodeIDs. The FSPNodeIndex of the access terminal is then given by the FSPNodeIndex of the base node with the ackable node index of $\min(m, N_{\text{ACKABLE}} - 1)$. The access network shall ensure that the access terminal assignments are such that the above rule assigns different FSPNodeIndex values to different access terminals on a given PHY frame.

To transmit an SP indication to an access terminal, the access network determines the FSPValue corresponding to the access terminal's FSPNodeIndex. If there is no SP indication for a base node, the FSPValue for that base node shall be set to 0.

The access network shall also determine the power (FSPPower) used to transmit each of these values, for use by the Physical Layer. Algorithms for determining this power are outside the scope of this standard.

The transmission of all this data to the PHY Layer is done in the following way:

- 1) Initialize an index k to 0.
- 2) If k is the FSPNodeIndex of any access terminal, then determine the FSPValue corresponding to the assignment of that terminal. Otherwise, set the FSPValue of this base node to 0.
- 3) If FSPValue $\neq$ 0
 - Determine FSPPower corresponding to the access terminal with FSPNodeIndex of k.
 - Provide the Physical Layer Protocol with the FSPValue, FSPNodeIndex = k, and FSPPower.
- 4) Increment k by 1.
- 5) If k < TotalFSPIndices, go to step 2).

8.5.5.4.2 Access terminal requirements

The following requirements apply when this protocol is in the active state.

8.5.5.4.2.1 Access terminal requirements in case of outstanding access probe

If the access terminal has an outstanding access probe with AccessType = 0, where AccessType is public data of the AC MAC protocol, the access terminal is only required to demodulate F-SCCH blocks that are scrambled by the AccessSequenceID of the outstanding access probe (i.e., the access terminal only needs to look for access grants), and whose index (i,j) falls in the range $j = 0$ and $0 \leq i < 5$.

If an AccessGrant block is received, and all the following are true:

- The access terminal has no assigned MACID on the sector,

- The *AccessSequenceID* used to scramble the *AccessGrant* matches the *AccessSequenceID* in the public data of the Access Channel MAC Protocol, and
- The *PilotID* of the sector that transmitted the *AccessGrant* matches the *PilotID* in the public data of the Access Channel MAC Protocol,

the access terminal shall

- Return an *AccessGrantReceived* indication specifying the applicable sector.
- Place the received *AccessGrant* in the public data.
- In the Physical Layer, adjust the timing for the sector that transmitted the access grant. In particular, the access terminal shall advance its timing by $\text{TimingAdjust} \times N_{\text{FFT}}/128$ chips if *TimingAdjust* is positive and shall retard its timing by $\text{abs}(\text{TimingAdjust}) \times N_{\text{FFT}}/128$ chips if *TimingAdjust* is negative. Here, *TimingAdjust* refers to the *Timing* field of the received *AccessGrant*, interpreted as a signed integer.
- Instruct the RCC MAC to initialize the transmit power as described in 8.7.5.4.6.
- Place the quantity *RPICHScramblingSeed* in its public data. *RPICHScramblingSeed* is defined in 8.5.5.4.2.1.1
- Issue a *ReverseControlChannelMAC.ACKAssignment* for RL PHY frame k_a , where $k_a = k + 3$ if *DuplexingMode* = FDD, $k_a = k$ if *DuplexingMode* = TDD44, and $k_a = k/2$ if *DuplexingMode* = TDD63, along with the index *i* if the *AccessGrant* block is received on FL PHY frame *k* and has an index pair (as defined in 8.5.5.4.1.1.3.6) equal to (*i*,0).
- Starting three FL PHY frames after the FL PHY frame in which the access grant is received, monitor the FLCS of the sector from which the access grant was received for F-SCCH blocks scrambled with its MACID or MACID_{BROADCAST}, as well as relevant instances F-PCCH, F-PQICH, F-FOSICH, F-IOTCH, F-ACKCH, F-SPCH, and F-RABCH.

If the access terminal has an outstanding access probe with *AccessType* = 1, where *AccessType* is public data of the AC MAC protocol, the access terminal is only required to demodulate F-SCCH blocks sent by the target sector of the access probe that are scrambled by the MACID of the access terminal in this sector. The set of F-SCCH blocks that are to be decoded by the access terminal are determined according to the following procedure:

- a) The access terminal shall decode F-SCCH blocks (*i*,*j*) with the value of *i* in the range 0, ..., $\min(\text{MaxNumSharedLABs}, \text{MaxSCCHDecodedBlocks}) - 1$ (indices of shared LABs).
- b) If $\text{MaxSCCHDecodedBlocks} > \text{MaxNumSharedLABs}$, define *S*(MACID) according to the following procedure:
 - 1) Denote $\text{MaxNumUnsharedLABs} = \text{MaxNumQPSKLABs} - \text{MaxNumSharedLABs}$.
 - 2) Denote $b = f_{\text{PHY-HASH}}(\text{MACID}) \bmod \text{MaxNumUnsharedLABs}$.
 - 3) Denote $x = \min(\text{MaxSCCHDecodedBlocks} - \text{MaxNumSharedLABs}, \text{MaxNumUnsharedLABs})$.
 - 4) Define L_k to be the total number of LABs excluding shared LABs that are contained in LAB segments with indices *k* or less (Here, the common segment has index 0 and the first LAB segment has index 1.) The Physical Layer Protocol specifies the notion of Common Segment and LAB Segment and the number of LABs in each segment. Denote by *s* the maximum integer that satisfies $L_{s-1} < b$.
 - 5) If $b + x - 1 < L_s$, then define *S*(MACID) = {*b*, ..., *b* + *x* - 1}.
 - 6) If $x \leq L_s - L_{s-1}$ and $b + x - 1 \geq L_s$, then define *S*(MACID) as follows:
For odd values of MACID, *S*(MACID) = { L_{s-1} , ..., $L_{s-1} + x - 1$ }. Else, *S*(MACID) = { $L_s - x$, ..., $L_s - 1$ }.
 - 7) If $x > L_s - L_{s-1}$ and $b + x - 1 < \text{MaxNumUnsharedLABs}$, then define *S*(MACID) = {*b*, ..., *b* + *x* - 1}.

- 8) If $x > L_s - L_{s-1}$ and $b + x - 1 \geq \text{MaxNumUnsharedLABs}$, then $S(\text{MACID}) = \{b, \dots, \text{MaxNumUnsharedLABs} - 1\} \cup \{0, \dots, x - 1 - (\text{MaxNumUnsharedLABs} - b)\}$.
- 9) The access terminal shall decode F-SCCH blocks $(\text{MaxNumSharedLABs} + i, j)$ with the value of i belonging to $S(\text{MACID})$.

For each value of i that is determined above, the access terminal shall decode F-SCCH blocks under two hypotheses if the following two conditions are satisfied:

- $i < \text{Max16QAMSCCHBlocks}$
- The access terminal has the attribute `HighSCCHModulationSupported` set to 1.

The two hypotheses are as follows:

- The blocks $(i, 0)$ and $(i, 1)$ are transmitted with 16-QAM modulation.
- The block $(i, 0)$ is transmitted with QPSK modulation.

If the above conditions are not satisfied, the access terminal shall decode the F-SCCH block only under the assumption that it is transmitted using QPSK modulation.

If an `AccessGrant` block is received and all the following are true

- The MACID in the `AccessGrant` matches the MACID assigned to the access terminal on the sector,
- The PilotID of the sector that transmitted the `AccessGrant` matches the PilotID in the public data of the Access Channel MAC Protocol,

the access terminal shall

- Return an *AccessGrantReceived* indication specifying the applicable sector.
- Place the received `AccessGrant` in the public data.
- Issue a `ReverseControlChannelMAC.ChangeFLSS` command to change the FLSS to the sector from which the access grant was sent, and also issue a `ReverseControlChannelMAC.ChangeRLSS` command to change the RLSS to the sector from which the access grant was sent.
- In the Physical Layer, adjust the timing for the sector that transmitted the access grant. In particular, the access terminal shall advance its timing by $\text{TimingAdjust} \times N_{\text{FFT}}/128$ chips if `TimingAdjust` is positive, and shall retard its timing by $\text{abs}(\text{TimingAdjust}) \times N_{\text{FFT}}/128$ chips if `TimingAdjust` is negative. Here, `TimingAdjust` refers to the `Timing` field of the received `AccessGrant`, interpreted as a signed integer.
- Instruct the RCC MAC to initialize the transmit power as described in 8.7.5.4.6.
- Issue a *ReverseControlChannelMAC.ACKAssignment* for RL PHY frame $k + 3$ along with the index i if the `AccessGrant` block is received on FL PHY frame k and has an index pair (as defined in 8.5.5.4.1.1.3.6) equal to $(i, 0)$.
- Starting three FL PHY frames after the FL PHY frame in which the access grant is received, monitor the FLCS of the sector from which the access grant was received for F-SCCH blocks scrambled with its MACID or `MACIDBROADCAST`, as well as relevant instances of F-PCCH, F-PQICH, F-FOSICH, F-IOTCH, F-ACKCH, F-SPCH, and F-RABCH.

All F-SCCH blocks other than `AccessGrant` blocks shall be discarded.

8.5.5.4.2.1.1 RPICHScramblingSeed

The RPICH scrambling seed is defined as shown in Table 96 and the text that follows.

Table 96—Format for RPICHScramblingSeed

Field	Length (bits)
PilotID	10
AccessSequenceID	10
MACID	11
FrameIndexLSBs	18

PilotID	The PilotID of the sector that sent the access grant. The notion of PilotID is defined in the Physical Layer Protocol (see 9.2.2.2.3).
AccessSequenceID	The AccessSequenceID in response to which the AccessGrant was received.
MACID	The MACID assigned in the AccessGrant.
FrameIndexLSBs	The 18 LSBs of the FL PHY frame index, starting from the beginning of system time, in the FL PHY frame where the AccessGrant was received.

8.5.5.4.2.2 Access terminal requirements in case of no outstanding access probe

If the access terminal has no outstanding access probe, then the access terminal shall decode relevant F-SCCH blocks scrambled with its MACID, as well as relevant instances of F-PCCH, F-PQICH, F-FOSICH, F-IOTCH, F-ACKCH, F-SPCH, and F-RABCH from the corresponding members of its active set. The notion of FLSS, DFLSS, RLSS, and DRLSS are defined by the RCC MAC Protocol (see 8.7).

Note that the access terminal is not required to demodulate any FLCS transmissions until three FL PHY frames after receiving any access grant.

If the access terminal has its parameter TuneAwayStatus in the public data of the Connected State Protocol set to 'I', the access terminal need not monitor any channel in the FLCS.

The following is a potential error event and specified action:

- Each block that does not correctly pass CRC shall be discarded.

Let each F-SCCH block be indexed with a pair (i,j) as described in 8.5.5.4.1.1.3.6.

If the access terminal has no assigned MACID from this sector, then the access terminal shall decode F-SCCH blocks (i,j) with $0 \leq i < \text{MaxSCCHDecodedBlocks}$.

If the access terminal has an assigned MACID from this sector, the access terminal shall decode F-SCCH blocks (i,j) for the values of i determined according to the following procedure:

- a) The access terminal shall decode F-SCCH blocks (i,j) with the value of i in the range 0, ..., $\min(\text{MaxNumSharedLABs}, \text{MaxSCCHDecodedBlocks}) - 1$.
- b) If $\text{MaxSCCHDecodedBlocks} > \text{MaxNumSharedLABs}$, define S(MACID) according to the following procedure:
 - Denote $\text{MaxNumUnsharedLABs} = \text{MaxNumQPSKLABs} - \text{MaxNumSharedLABs}$.

- Denote $b = f_{\text{PHY-HASH}}(\text{MACID}) \bmod \text{MaxNumUnsharedLABs}$.
- Denote $x = \min(\text{MaxSCCHDecodedBlocks} - \text{MaxNumSharedLABs}, \text{MaxNumUnsharedLABs})$.
- Define L_k to be the total number of LABs, excluding shared LABs, that are contained in LAB segments with indices k or less. (Here, the common segment has index 0 and the first LAB segment has index 1.) The Physical Layer Protocol specifies the notion of Common Segment and LAB Segment and the number of LABs in each segment. Denote by s the maximum integer that satisfies $L_{s-1} < b$.
- If $b + x - 1 < L_s$, then define $S(\text{MACID}) = \{b, \dots, b + x - 1\}$.
- If $x \leq L_s - L_{s-1}$ and $b + x \geq L_s$, then define $S(\text{MACID})$ as follows:
For odd values of MACID , $S(\text{MACID}) = \{L_{s-1}, \dots, L_{s-1} + x - 1\}$. Else, $S(\text{MACID}) = \{L_s - x, \dots, L_s - 1\}$.
- If $x > L_s - L_{s-1}$ and $b + x - 1 < \text{MaxNumUnsharedLABs}$, then define $S(\text{MACID}) = \{b, \dots, b + x - 1\}$.
- If $x > L_s - L_{s-1}$ and $b + x - 1 \geq \text{MaxNumUnsharedLABs}$, then $S(\text{MACID}) = \{b, \dots, \text{MaxNumUnsharedLABs} - 1\} \cup \{0, 1, \dots, x - 1 - (\text{MaxNumUnsharedSharedLABs} - b)\}$.
- The access terminal shall decode F-SCCH blocks ($\text{MaxNumUnsharedLABs} + i_j$) with the value of i belonging to $S(\text{MACID})$.

For each value of i that is determined above, the access terminal shall decode F-SCCH blocks under two hypotheses if the following two conditions are satisfied:

- $i < \text{Max16QAMSCCHBlocks}$
- The access terminal has the attribute `HighSCCHModulationSupported` set to 1.

The two hypotheses are as follows:

- The blocks $(i,0)$ and $(i,1)$ are transmitted with 16-QAM modulation.
- The block $(i,0)$ is transmitted with QPSK modulation.

If the above conditions are not satisfied, the access terminal shall decode the F-SCCH block only under the assumption that it is transmitted using QPSK modulation.

The access terminal shall perform the following actions for decoding F-SCCH blocks:

- If an AccessGrant block is received and the MACID in the AccessGrant matches the MACID assigned to the access terminal on the sector, the access terminal shall adjust the timing for the sector that transmitted the access grant in the Physical Layer. In particular, the access terminal shall advance its timing by $\text{TimingAdjust} \times N_{\text{FFT}}/128$ chips if `TimingAdjust` is positive and shall retard its timing by $\text{abs}(\text{TimingAdjust}) \times N_{\text{FFT}}/128$ chips if `TimingAdjust` is negative. Here, `TimingAdjust` refers to the `Timing` field of the received AccessGrant, interpreted as a signed integer. The access terminal shall also issue a *ReverseControlChannelMAC.ACKAssignment* for RL PHY frame $k + 3$ along with the index i if the AccessGrant block is received on FL PHY frame k and has an index pair (as defined in 8.5.5.4.1.1.3.6) equal to $(i,0)$.
- If an F-SCCH block that was scrambled with $\text{MACID}_{\text{PAGE}}$ is received, then the block can be TD-FLAB.
- If an F-SCCH block that was scrambled with $\text{MACID}_{\text{BROADCAST}}$ is received, then the block can be one of PDCAB, TD-FLAB, or GRA and the block type is determined by the block header.
- If an F-SCCH block that was scrambled with an individual MACID is received, then that block is not a PDCAB or a GRA block. The block type is determined by the variable `MIMOMode`. If `MIMOMode = SCW`, then, if there are multiple block types 8.5.5.4.1.1.1 with the same block header (where block types and block headers are as described in 8.5.5.4.1.1.1, the received block is a SCW

block. If MIMOMode = MCW, then, if there are multiple block types 8.5.5.4.1.1.1 with the same block header, the received block is a MCW block.

- If an MCW block with block header '110' is received, then the access terminal shall determine the block type based on the field BlockType according to the description of this field.
- If a TD-FLAB, MCW-FLAB1, MCW-FLAB2, SCW-FLAB, SZ-TD-FLAB, SZ-MCW-FLAB1, SZ-SCW-FLAB, RRA-TD-FLAB, RRA-MCW-FLAB1, or RRA-SCW-FLAB is received, the access terminal shall invoke the procedures for access terminal assignment management in the FTC MAC Protocol.
- If an RLAB is received, the access terminal shall invoke the procedures for access terminal assignment management in the RTC MAC Protocol.

The access terminal shall perform the following actions on the remaining blocks/bits:

- For any RL assignment (RL-ATA or RL-NP-ATA as defined by the RTC MAC Protocol), the access terminal shall monitor the corresponding F-ACKCH transmissions and perform the actions contingent upon the reception of explicit (values 1, 2, or 3) or implicit (0) F-ACKCH values as specified in the RTC MAC Protocol.
- For any persistent FL assignment (FL-ATA as defined by the FTC MAC Protocol), the access terminal shall monitor the corresponding F-SPCH transmissions and perform the actions contingent upon the reception of explicit (values 1, 2, or 3) or implicit (0) F-SPCH values as specified in the FTC MAC Protocol.
- The access terminal shall determine its transmit power based on its power control bits received on F-PCCH from its RLSS as specified in the RCC MAC Protocol.
- The access terminal shall determine the pCoT values based on the F-PQICH reports from active set members as described in the RCC MAC Protocol.
- For each sector in the active set, if FastOSIEnabled is set to '1', the access terminal shall monitor the relevant Fast OSI values and provide the received values to the Reverse Traffic Channel MAC Protocol.
- For each sector in the active set, if IoTEnabled is set to '1', the access terminal shall monitor the relevant F-IOTCH values and provide the received values to the Reverse Traffic Channel MAC and Reverse Control Channel MAC Protocols.

8.5.5.4.3 General procedures

8.5.5.4.3.1 Procedures for half-duplex mode

8.5.5.4.3.1.1 Access network procedures

If the HalfDuplexMode bit in the public data of the Active State Management Protocol is equal to one, then the following are true:

- If the MACID assigned to the terminal is even, then the access network shall not send any forward link assignment blocks or reverse link assignment blocks to the access terminal on odd-indexed frames. Additionally, the access network shall not transmit any link assignments with the ExtendedTransmission field set to 1.
- If the MACID assigned to the terminal is odd, then the access network shall not send any forward link assignment blocks or reverse link assignment blocks to the access terminal on even-indexed frames. Additionally, the access network shall not transmit any link assignments with the ExtendedTransmission field set to 1.
- If the access terminal has no assigned MACID, the access network may send AccessGrant blocks to the access terminal on any FL PHY frame subject to the constraints specified in the AC MAC protocol.

8.5.5.4.3.1.2 Access terminal procedures

If the HalfDuplexMode bit in the public data of the Active State Management Protocol is equal to one, then the following are true:

- If the MACID assigned to the terminal is even, then the access terminal shall not monitor any forward link assignment blocks or reverse link assignment blocks on odd-indexed frames. Additionally, the access terminal is not required to monitor F-IOTCH reports that are transmitted on odd-indexed frames and shall treat them as erasures when determining the corresponding interference offset values as described in 8.7.5.4.6.2.
- If the MACID assigned to the terminal is odd, then the access terminal shall ignore any forward link assignment blocks or reverse link assignment blocks on even-indexed frames. Additionally, the access terminal is not required to monitor F-IOTCH reports that are transmitted on even-indexed frames and shall treat them as erasures when determining the corresponding interference offset values as described in 8.7.5.4.6.2.
In both cases above, the access terminal shall ignore any link assignments with the ExtendedTransmission field set to 1.
- If the access terminal has no assigned MACID, the access terminal shall monitor all relevant FL PHY frames, as specified in the AC MAC protocol, for AccessGrant blocks.

8.5.5.4.3.2 Procedures for SelectedInterlace Operation

8.5.5.4.3.2.1 Access network procedures

If the SelectedInterlaceMode public data of the Connected State Protocol is equal to one, and the SelectedInterlaceAssignment public data of the Connected State Protocol includes the PilotID of the current serving sector, then the access network shall not send any forward link assignment blocks or reverse link assignment blocks to the access terminal on any FL PHY frame that is not part of the SelectedInterlaceSet. Additionally, the access network shall not transmit any forward link assignments with the ExtendedTransmission field set to 1.

8.5.5.4.3.2.2 Access terminal procedures

If the SelectedInterlaceMode public data of the Connected State Protocol is equal to one, and the SelectedInterlaceAssignment public data of the Connected State Protocol includes the PilotID of the current serving sector, then the access terminal may stop monitoring F SCCH (for FLABs or RLABs) and F-IOTCH on interlaces that are not part of the SelectedInterlaceSet. Additionally, the access terminal shall ignore any forward link assignments with the ExtendedTransmission field set to 1.

8.5.5.4.3.3 Procedures for Semi-Connected State

The access terminal shall perform the following steps each upon receipt of an FLAB from the SCCH-MAC Protocol when it is in semi-connected state (as determined by the state of the Connected State Protocol). If any FLAB with NodeID NodeID_{DEASSIGN} and NULL packet format is received, the access terminal shall return a *SemiConnectedExitReceived* indication.

8.5.5.4.3.4 Procedures for TuneAway Operation

If the access terminal has its parameter TuneAwayStatus in the public data of the Air Link Management Protocol set to '1', the access terminal need not monitor any F-SCCH transmissions.

8.5.6 Message formats for the InUse instance

8.5.6.1 TimingCorrection

The access network shall send the timing correction message to correct the reverse link timing of the access terminal (see Table 97).

Table 97—TimingCorrection message

Field	Length (bits)
MessageID	8
TimingCorrection	16
PilotID	10
Reserved	6

MessageID	The access network shall set this field to 0x00.
TimingCorrection	The access network shall set this field to the timing correction on the sector in twos complement format in units of 1/8 chips. A positive value shall advance the timing and a negative value shall retard the timing.
PilotID	The access network shall set this field to the PilotID of the sector.
Reserved	The receiver shall ignore this field.

Channels	FTC	RLP	Best Effort
Addressing	Unicast	AuthTag	Required when key is available

8.5.7 Header and trailer formats

This protocol does not carry higher-layer payload, and thus header and trailer formats are not defined.

8.5.8 Interface to other protocols

8.5.8.1 Commands

This protocol does not issue any commands.

8.5.8.2 Indications

This protocol registers to receive the following indication:

- *AccessChannelMAC.AccessProbeReceived*

8.5.9 Configuration attributes

8.5.9.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 98. The access terminal shall use as defaults the values in Table 98 that are listed in ***bold italics***.

Table 98—Configurable simple attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	MaxSCCHDecodedBlocks Capability: access network shall not propose this attribute	Soft Allowed/Static	<i>0x05</i>	The total number of F-SCCH blocks the access terminal is able to decode from each sector it monitors.
			0x06–0x20	The total number of F-SCCH blocks the access terminal is able to decode from each sector it monitors.
			All other values	Reserved.
0x0001	PDCABSupported Capability: access network shall not propose this attribute	Soft Allowed/Static	<i>0x00</i>	The access terminal does not support PDCAB.
			0x01	The access terminal supports PDCAB.
			All other values	Reserved.
0x0002	HighSCCHModulationSupported Capability: access network shall not propose this attribute	Soft Allowed/Static	<i>0x00</i>	The access terminal does not support high modulation on F-SCCH.
			0x01	The access terminal supports high modulation on F-SCCH. If the access terminal sets MaxMCWLayers (FTC MAC) to a value of two or more, the access terminal shall support high modulation on F-SCCH.
			All other values	Reserved.
0x0003	SCCHTailBitingCodeSupported Capability: access network shall not propose this attribute	Soft Allowed/Static	<i>0x00</i>	The access terminal does not support tail-biting code for F-SCCH.
			0x01	The access terminal supports tail-biting code for F-SCCH
			All other values	Reserved.

8.5.10 Protocol numeric constants

Protocol numeric constants are listed in Table 99.

Table 99—Numeric constants of the Forward Link Control Segment MAC Protocol

Constant	Meaning	Value
N _{FLSCMPType}	Type field for this protocol	See 16.1
N _{FLSCMPDefault}	Subtype field for this protocol	0x00
MACID _{BROADCAST}	Broadcast MACID	0
MACID _{PAGE}	Broadcast MACID, used for paging	1

8.5.11 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

8.6 Basic Forward Traffic Channel MAC Protocol

8.6.1 Overview

The Basic Forward Traffic Channel MAC Protocol provides the procedures and messages required for an access network to transmit, and an access terminal to receive, the Forward Traffic Channel.

The access network maintains an instance of this protocol for every assigned MACID.

This protocol at the access terminal operates in one of the following two states:

- *Inactive State*: In this state, the access terminal cannot receive any packets on the Forward Traffic Channel. When the protocol is in this state, it waits for an *Activate* command.
- *Active State*: In this state, the access terminal receives the forward traffic channel based on link assignment blocks received from the FLCS MAC Protocol.

The protocol states and allowed transitions between the states are shown in Figure 56. The rules governing these transitions are provided in 8.6.5.2.1 and 8.6.5.2.2.

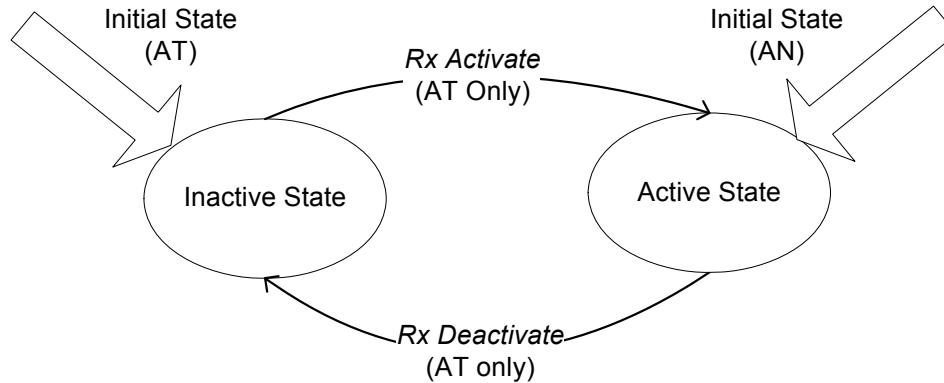


Figure 56—State Diagram for Forward Traffic Channel MAC Protocol

This protocol uses the parameters and attributes listed in Table 100, where OMP and ASMP refer to the Overhead Messages Protocol and Active Set Management Protocol, respectively.

Table 100—Parameters and attributes for the Forward Traffic Channel MAC Protocol

Parameter Name	Where defined	Comments
AssignmentAckHARQTransmission	OMP	ExtendedChannelInfo
DFLSS	RCC MAC Protocol	Public data
DuplexingMode	OMP	Public data
FLNumSDMADimensions	OMP	QuickChannelInfo
FLSS	RCC MAC Protocol	Public data
FLSubzoneSize	OMP	QuickChannelInfo
LDPCSupportedFL	ASMP	Public data
$N_{CRC,Data}$	PHY	Public data
NumDRCHSubzones	OMP	QuickChannelInfo
NumFailPacketsforDrop	ASMP	Public data
PagingIDList	RouteControlLayer	PublicData
PDCABResourceSharingEnabled	OMP	ExtendedChannelInfo
PDCABSupported	Configuration attribute	FLCS MAC Protocol
ResourceChannelMuxMode	OMP	QuickChannelInfo
StartofPacketFrame	ASMP	Public data
StartofPacketMode	ASMP	Public data
TotalNumSubcarriers	OMP	SystemInfo
TuneAwayStatus	ASMP	Public data

8.6.2 Primitives

8.6.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

8.6.2.2 Return indications

This protocol returns the following indications:

- *forwardtrafficpacketsmissed*
- *supervisionfailed*
- *PageReceived* (with *PageResponseAccessSequence* and *AccessPersistenceClass* optionally included)
- *SemiConnectedExitReceived*

8.6.2.3 Local common data

This protocol does not define any local common data.

8.6.2.4 Public data

This protocol shall make the following data public:

Subtype for this protocol.

8.6.3 Protocol Data Unit

The transmission unit of this protocol is a Forward Traffic Channel MAC layer packet.

8.6.4 Procedures and messages for the InConfiguration instance of the protocol

8.6.4.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

8.6.4.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

8.6.4.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

8.6.5 Procedures for the InUse instance

This protocol constructs a Forward Traffic Channel MAC packet out of the Consolidated Packet received from the Packet Consolidation Protocol. The protocol then sends the packet for transmission to the Physical Layer.

8.6.5.1 Initialization and commit procedures

8.6.5.1.1 Protocol initialization for the InUse Protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the InActive State.
- The protocol at the access network shall enter the Active State.

8.6.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

8.6.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

8.6.5.2 Command processing

8.6.5.2.1 Activate

If this protocol receives an *Activate* command in the Inactive State, the access terminal and the access network shall transition to the Active State.

If this protocol receives the command in any other state, the command shall be ignored.

8.6.5.2.2 Deactivate

If the protocol receives a *Deactivate* command in the Active State,

- The access terminal shall cease monitoring the Forward Traffic Channel and shall transition to the Inactive State.
- The access network shall ignore the command.

If this command is received in the Inactive State, the command shall be ignored.

8.6.5.3 Forward Traffic Channel Addressing

Transmission on the Forward Traffic Channel is multiplexed in time and frequency. An assignment on the Forward Traffic Channel is specified by a set of hop-ports (frequency) and a set of PHY frames in time. Each hop-port is specified by a hop-port index. Packets transmitted over the Forward Traffic Channel are transmitted over the F-DCH Physical Layer channel. F-DCH is the primary channel for user data transmission, and access terminals are assigned F-DCH resources (i.e., time-frequency resources) via assignment blocks that are sent over the F-SCCH. Note that the assignment blocks MCW-FLAB2 and RRA-MCW-FLAB1 are indistinguishable when they are passed from the F-SCCH to the FTC-MAC, and the first field, the block type field, must be parsed for differentiation.

The duration of an assignment of hop-ports may or may not be pre-specified. Assignments whose durations are pre-specified are known as non-persistent assignments, and assignments whose durations are not pre-specified are known as persistent assignments. Non-persistent assignments may be unicast, or they may be

broadcast. Residual assignments, defined below, have pre-specified durations, but are not included in the category of non-persistent assignments. An access terminal may have the following types of assignments:

- Persistent assignments: denoted by FL-ATA
- Non-persistent assignments unicast: denoted by FL-NP-ATA_{UC}
- Non-persistent broadcast assignments: FL-NP-ATA_{BC}
- Residual-resource assignments: FL-R-ATA
- Group resource assignments: FL-GR-ATA

These types of assignments are explained in more detail below.

FL-ATA: The set of hop-ports assigned for a particular set of PHY frames whose duration is not pre-specified is referred to as a “Forward Link Access Terminal Assignment” or FL-ATA. The FL-ATA is generally assigned via TD-FLABs/SCW-FLABs/MCW-FLABs that have the persistent bit set to '1'; the exception being that a FL-ATA may be altered for a specified length of time via a TD-FLAB/SCW-FLAB/MCW-FLAB with the persistent bit set to '0'. (See 8.6.5.5.1.1.3.2.1 for more details.) These TD-FLABs/SCW-FLABs/MCW-FLABs are scrambled with the access terminal’s MACID. A FLATA for a particular set of PHY frames may only be assigned via a single type of assignment block. (The three types of FL assignment blocks are TDFLABs, SCW-FLABs, and MCW-FLABs. See 8.5.5.4.1.1.1 for complete definitions of these block types.) A FL-ATA that has been assigned via a particular type of assignment block is also associated with that type. More precisely, a FL-ATA may be of type TDFLAB, SCW-FLAB, or MCW-FLAB. The access terminal may have multiple FL-ATAs, as long as they do not overlap in time. (The concept of time overlapping ATAs is discussed in 8.6.5.3.1.)

FL-NP-ATA_{UC}: The set of hop-ports assigned for a particular set of PHY frames whose duration is pre-specified is referred to as a “Forward Link Non-Persistent Access Terminal Unicast Assignment” or FL-NP-ATA_{UC}. The FL-NP-ATA_{UC} is assigned via TD-FLABs/SCW-FLABs/MCW-FLABs that have the persistent bit set to '0', or SZ-TD-FLABs/SZ-MCW-FLABs/SZ-SCW-FLABs that are scrambled with the access terminal’s MACID. These aforementioned FLABs are sometimes referred to as NP-FLAB_{UCS}. A FL-NP-ATA_{UC} may only be assigned via a single type of assignment block (TD-FLAB, SCW-FLAB, MCW-FLAB, SZ-TDFLAB, SZ-MCW-FLAB, or SZ-SCW-FLAB). A FLNPATA_{UC} that has been assigned via a particular type of assignment block is also associated with that type. More precisely, a FL-NP-ATA_{UC} may be of type TD-FLAB, SCW-FLAB, MCW-FLAB, SZ-TD-FLAB, SZ-MCW-FLAB, or SZ-SCW-FLAB. An access terminal may have multiple FL-NP-ATA_{UCS}, as long as they do not overlap in time.

FL-NPATA_{BC}: The set of hop-ports assigned for broadcast for a particular set of PHY frames, whose duration is pre-specified is referred to as a “Forward Link Non-Persistent Access Terminal Broadcast Assignment” or FLNP-ATA_{BC}. The FL-NP-ATA_{BC} is assigned via TD-FLABs that have the persistent bit set to '0', and are scrambled with the broadcast MACID. These FLABs are denoted NP-FLAB_{BCS}. Non-persistent broadcast assignment blocks include only TDFLABs. An access terminal may have multiple FL-NP-ATA_{BCS}, as long as they do not overlap in time. The broadcast MACID is given by MACID_{BROADCAST}, which is a constant of the FLCS MAC protocol.

FL-R-ATA: The set of hop-ports assigned for a particular access terminal for a particular set of PHY frames via RRA-TD-FLABs/RRA-MCW-FLABs/RRA-SCW-FLABs (and the GRA bitmap) is known as a “Forward Link Residual Access Terminal Assignment” or FL-R-ATA. RRA-TD-FLABs/RRA-MCW-FLABs/RRA-SCW-FLABs are sometimes referred to as RRA-FLABs. A FL-R-ATA may only be assigned via a single type of assignment block (RRA-TD-FLAB, RRA-MCW-FLABs, or RRA-SCW-FLABs). A FLRATA that has been assigned via a particular type of assignment block is also associated with that type. More precisely, a FL-R-ATA may be of type RRA-TD-FLAB, RRA-MCW-FLABs, or RRA-SCW-FLABs. An access terminal may have multiple FL-R-ATAs, as long as they do not overlap in time.

The access network shall assign the access terminal to a group using a Group Assignment Message (GAM) prior to assigning an FL-R-ATA to the access terminal through an RRA-FLAB. If the access terminal has an FL-R-ATA, it shall monitor one or more GRA bitmaps as indicated in the GAM to determine its final assignment.

FL-GR-ATA: The access terminal may be assigned one or more sets of hop-ports using the GRA feature. This ATA is referred to as a “Forward Link Group Resource Access Terminal Assignment” and is managed using the GAM and a group resource allocation bitmap that is transmitted on the F-DCH or F-SCCH.

Sets of hop-ports assigned in assignment blocks received over the F-SCCH are specified using the channel tree. The channel tree is as specified in 8.6.5.6.

The access network shall assign the access terminal to a group using a GAM prior to assigning an FL-GR-ATA to the access terminal. If the access terminal has an FL-GR-ATA, it shall monitor one or more GRA bitmaps as indicated in the GAM to determine its final assignment.

8.6.5.3.1 Time overlapping assignments

The PHY frames used for an FL-ATA/FL-NP-ATA/FL-R-ATA depend on which PHY frame the original assignment block assigning the FL-ATA/FL-NP-ATA/FL-R-ATA was received, as well as the values of the extended transmission field and HARQ field in the assignment block/blocks. The HARQ field specifies which timing structure is used, and the extended transmission field specified whether or not extended transmissions are used. (See 8.1.5.1 for the PHY frames that would correspond to the various timing structures with and without extended transmissions.) For the purposes of determining whether an FL-ATA/FL-NP-ATA/FL-R-ATA overlaps in time with another FL-ATA/FL-NP-ATA/FL-R-ATA, it is assumed that a FL-ATA will last infinitely into the future, and that a FL-NP-ATA/FL-R-ATA will last for the maximum number of HARQ retransmissions. If the intersection of the set of PHY frames for two ATAs is non-empty, then those two ATAs are considered to be time overlapping. As an example, consider FL-ATA1 assigned on PHY frame 1 that has a timing structure with transmissions every six FL-PHY frames, and no extended transmission, and FL-ATA2 that is assigned on PHY frame 3 and has a timing structure with transmissions spaced every eight FL-PHY frames apart, and no extended transmission. The set of PHY frames for FL-ATA1 is $1 + 6 \times n$, where $n > 0$. The set of PHY frames for FL-ATA2 is $3 + 8 \times m$ for $m > 0$. Since the intersection of these PHY frames is non-empty, the FL-ATA1 and FL-ATA2 overlap in time.

8.6.5.3.2 Summary of Rules for Forward Traffic Channel Addressing

As a consequence of the rules in this subclause regarding the coexistence of FL-ATAs/FL-NP-ATA_{BC}s/FL-NP-ATA_{UC}s the following applies for a given access terminal:

- No two FL-ATAs can overlap in time; no two FL-NP-ATA_{BC}s can overlap in time; no two FL-NP-ATA_{UC}s can overlap in time; and no FL-R-ATAs can overlap in time.
- A FL-ATA and a FL-NP-ATA_{UC} cannot overlap in time.
- A FL-ATA and a FL-R-ATA cannot overlap in time.
- A FL-NP-ATA_{UC} and a FL-R-ATA cannot overlap in time.
- A FL-NP-ATA_{BC} cannot simultaneously overlap in both time and frequency with a FL-ATA, a FL-NP-ATA_{UC}, or a FL-R-ATA.

The access network shall ensure that the following rules hold:

- A FL-ATA/FL-NP-ATA_{BC}/ FL-NP-ATA_{UC}s/ FL-R-ATA/ FL-GR-ATA shall not span multiple SDMA subtrees. (See 8.6.5.6 for definition of SDMA subtree.) Additionally, an FL-NP-ATA_{BC} shall not span any SDMA subtree other than subtree 0, while an FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA/FL-GR-ATA shall not span any subtree with index larger than MaxNumFLSDMASubtrees.

- A FL-ATA/ FL-NP-ATA_{BC}/ FL-NP-ATA_{UC}/ FL-R-ATA/FL-GR-ATA shall not simultaneously include hop-ports that belong to the DRCH zone and hop-ports that belong to the BRCH zone. (8.6.5.6 defines the set of hop-ports that are assigned to BRCH and DRCH zones.)
- A FL-ATA/ FL-NP-ATA_{BC}/ FL-NP-ATA_{UC}/ FL-R-ATA/FL-GR-ATA shall not include hop-ports that are allocated to a BCMCS if the access terminal has the attribute SupercastSupported set to 0. The Physical Layer defines the set of hop-ports that are allocated to BCMCS.
- A FL-ATA/FL-NP-ATA_{BC}/FL-NP-ATA_{UC}/FL-R-ATA/FL-GR-ATA shall not simultaneously include hop-ports that are allocated to BCMCS segment and hop-ports that are not allocated to BCMCS segment.
- If DuplexingMode = FDD the packet size of a FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA/FL-GR-ATA shall not be greater than $\max(\text{MaxPacketSizeEightInterlace}, \text{MaxPacketSizeEightInterlaceLDPC}, \text{MaxPacketSizeEightInterlaceExtended}, \text{or } \max(\text{MaxPacketSizeSixInterlace}, \text{MaxPacketSizeSixInterlaceLDPC}))$, depending on the interlacing structure being used for this ATA. Additionally, if the ASMP parameter LDPCEnabledRL is set to '0', the packet size of a FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA/FL-GR-ATA shall not be greater than $\max(\text{MaxPacketSizeEightInterlace}, \text{MaxPacketSizeEightInterlaceExtended}, \text{or } \text{MaxPacketSizeSixInterlace})$, depending on the interlacing structure being used for this ATA. The packet size of a FL-NP-ATA_{BC} shall not be larger than $N_{\text{FTCMaXBCPacketSize}}$.
- If DuplexingMode = TDD63, all packet size restrictions refer to the size of any of the half-packets that is transmitted on a single FL PHY frame.
- For TDD, the packet size of a FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA/FL-GR-ATA shall not be greater than $\max(\text{MaxPacketSizeTDD44}, \text{MaxPacketSizeLDPCTDD44})$ if DuplexingMode = TDD44 or $\max(\text{MaxPacketSizeTDD63}, \text{MaxPacketSizeLDPCTDD63})$ if DuplexingMode = TDD63. Additionally, if the ASMP parameter LDPCSupportedFL is set to '0', the packet size of a FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA/FL-GR-ATA shall not be greater than $\max(\text{MaxPacketSizeTDD44}, \text{MaxPacketSizeTDD63})$ if DuplexingMode = TDD44, or $\text{MaxPacketSizeTDD63}$ if DuplexingMode = TDD63. The packet size of a FL-NP-ATA_{BC} shall not be larger than $N_{\text{FTCMaXBCPacketSize}}$.
- The number of usable hop-ports in a FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA/FL-GR-ATA shall not be greater than that indicated by the complex attribute FLSupportedAssignmentSize. Note that this attribute defines the maximum number of hop-ports allowable for different interlacing structures and MIMO configurations. If DuplexingMode = FDD, an FL-NP-ATA_{BC} shall always use the 8-interlace structure defined in 8.1.4. If DuplexingMode = TDD44, an FL-NP-ATABC shall always use the 4-interlace structure, which is defined in 8.1.6.1.1, and if DuplexingMode = TDD63, then, an FL-NP-ATABC shall always use the 3-interlace structure, which is defined in 8.1.7.1.1. The number of hop-ports in an FL-NP-ATA_{BC} shall not be larger than 512.
- The rank for a FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA of type SCW shall not be larger than MaxSCWStreams. The number of layers for a FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA of type MCW shall not be larger than MaxMCWLayers.
- An FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA/FL-GR-ATA shall not have the Pilot Format set to two if the AT has the attribute PilotFormat2Supported set to '0'. An FL-NP-ATA_{BC} shall not have the Pilot Format set to two.

All hop-ports in an FL-ATA/FL-NP-ATA_{UC}/ FLNP-ATA_{BC}/FL-R-ATA for a particular PHY frame shall be combined for transmission over the Physical Layer channel (F-DCH). Different FL-ATAs/FL-NP-ATAs/FL-R-ATAs shall always carry separate MAC packets with independent HARQ termination.

8.6.5.4 Inactive State

When the protocol is in the Inactive State, the access terminal waits for an *Activate* command.

The Inactive State is not defined for the access network.

8.6.5.5 Active State

In the Active State, the access network shall transmit over the F-DCH using the FL-ATAs /FL NP ATAs/ FL-R-ATAs/FL-GR-ATAs, and PFs selected by the access network and signaled to the access terminal over the F-SCCH Physical Layer channel or, in the case of FL-GR-ATAs, signaled via the group assignment message. The access terminal processes assignment blocks from the FLCS MAC protocol to maintain its FL-ATAs/FL-NP-ATAs/FL-R-ATAs and processes one or more GRA bitmaps from the FTC MAC protocol to maintain its FL-GR-ATAs. The access terminal configures the Physical Layer for reception of packets according to the ATAs. The access terminal controls transmission of ACK/NACK information via the RCC MAC Protocol based on pass or fail of the Lower MAC packet, as indicated by the PHY. The MACID assigned to the access terminal for each sector in its active set is given in the MACResourceAssignment message that is public data of the Active Set Management Protocol. The MACID is used to scramble LABs from the F-SCCH. Assignment and HARQ logic for the access terminal and access network are specified in 8.6.5.5.1 and 8.6.5.5.2, respectively.

8.6.5.5.1 Access terminal requirements

An access terminal shall keep a variable NumLayers for each FL-ATA that is of type SCW or MCW. On reception of a SCW-FLAB for a particular set of PHY frames from the FLCS MAC protocol that is not discarded due to assignment management rules, NumLayers for the resulting ATA shall be set according to the FL Pilot/MIMO field of the SCW-FLAB. On reception of a MCW-FLAB from the FLCS MAC protocol that is not discarded due to assignment management rules, NumLayers for the resulting ATA shall be set to the number of non-zero packet formats given in the MCW-FLAB.

8.6.5.5.1.1 Access terminal assignment management

8.6.5.5.1.1.1 Access terminal rules for processing assignment blocks

This subclause specifies the rules for interpreting and combining the forward link assignment blocks that are received in each FL PHY frame from the FLCS MAC protocol. These assignment blocks include those that are scrambled with the MACID of the access terminal as well as those that are scrambled with $\text{MACID}_{\text{BROADCAST}}$.

In this subclause, the term “FLAB” will be used to indicate all types of FL assignment blocks: TD-FLAB, MCW-FLAB, and SCW-FLAB, that have the persistent bit set to '1', unless otherwise specified. Similarly, the term “NP-FLAB” will be used to indicate FL assignment blocks: TD-FLAB, MCW-FLAB, SCW-FLAB, that have the persistent bit set to '0', as well as SZ-TD-FLAB, SZ-MCW-FLAB, and SZ-MCW-FLAB, unless otherwise specified. The term $\text{NP-FLAB}_{\text{UC}}$ will refer to NP-FLABs that are scrambled with the access terminal's MACID, while the term $\text{NP-FLAB}_{\text{BC}}$ will refer to NP-FLABs that are scrambled with $\text{MACID}_{\text{BROADCAST}}$. Finally, the term “RRA-FLAB” will be used to indicate FL assignment blocks RRA-TD-FLAB, RRA-SCW-FLAB, and RRA-MCW-FLAB.

In the following list, “x-FLAB” refers to any of the following: FLAB, $\text{NP-FLAB}_{\text{UC}}$, $\text{NP-FLAB}_{\text{BC}}$, or RRA-FLAB. Similarly, “x-ATA” refers to any of the following: FL-ATA, $\text{FL-NP-ATA}_{\text{UC}}$, $\text{FL-NP-ATA}_{\text{BC}}$, or FL-R-ATA. The following procedure for ordering and processing x-FLABs shall be applied separately to the set of FLABs, the set of $\text{NP-FLAB}_{\text{UC}}$ s, the set of $\text{NP-FLAB}_{\text{BC}}$ s, and the set of RRA-FLABs:

- a) If any x-FLAB received from the FLCS MAC protocol has the NodeID field set to $\text{NodeID}_{\text{DEASSIGN}}$, has the same type as the corresponding existing x-ATA (i.e., FL-ATA if the received x-FLAB is an FLAB, $\text{FL-NP-ATA}_{\text{UC}}$ if the received x-FLAB is an $\text{NP-FLAB}_{\text{UC}}$), and has the PF field set to the current packet format of the corresponding existing x-ATA, then this x-FLAB shall be treated as an x-FLAB for deassignment of the corresponding x-ATA. In addition, any other simultaneously received x-FLABs shall be discarded. Deassignment is described for FL-ATAs, $\text{FL-NP-ATA}_{\text{UC}}$ s, (unicast and broadcast) and FL-R-ATAs in 8.6.5.5.1.1.3.1, 8.6.5.5.1.1.4.1, and

8.6.5.5.1.1.5.1, respectively. The subsequently described steps in this subclause shall not be carried out in this case. Additionally, if an F-SPCH value of 2 is received from the FLCS MAC in this PHY frame, this shall be treated as a deassignment for FL-ATAs and FL-NP-ATA_{UCS}. Any other simultaneously received FLABs or NP-FLAB_{UCS} shall be discarded. In the case of an x-FLAB of type MCW, the condition only applies to the x-MCW-FLAB1 (and to the layer 1 PF contained in the MCW-FLAB1).

- b) If multiple x-FLABs are received from the FLCS MAC protocol in the same FL PHY frame, these x-FLABs are ordered according to the F-SCCH index pair (i,j) defined in 8.5.5.4.1.1.3.6. An x-FLAB (i,j) is said to have a lower index pair than an x-FLAB (i',j') if either $i < i'$, or $i = i'$ and $j < j'$. The set of hop-ports of the x-FLAB with the lowest index pair is denoted by N_0 , the set of hop-ports of the x-FLAB with the next higher index pair is denoted by N_1 , and so on. The set of hop-ports indexed by an x-FLAB is given either by the NodeID field or by the SZ field, depending on the type of x-FLAB. In the above, a NodeID equal to NodeID_{DEASSIGN} is mapped to the null set of hop-ports, denoted by Φ .
- c) If multiple x-FLABs are received from the FLCS MAC protocol in the same FL PHY frame and these x-FLABs are not all of the same type, the access terminal shall determine that an error has occurred. In this case, the access terminal shall ignore all these x-FLABs. Additionally, the remaining steps shown in this subclause shall not be carried out. Further, if all the received x-FLABs are of type MCW, and if any x-FLAB, except the one with the largest index, is an x-MCW-FLAB2, the access terminal shall determine that an error has occurred and shall ignore all these x-FLABs. Otherwise, the subsequent steps for interpreting the received x-FLABs shall be carried out only for the received x-MCW-FLAB1s. These x-MCW-FLAB1s are combined as described in the subsequent steps and are then passed to the appropriate procedure (Resource Reassignment, HARQ extension, non-supplemental assignment, supplemental assignment) as determined in these steps along with the received x-MCW-FLAB2, if any.
- d) If multiple x-FLABs are received from the FLCS MAC protocol in the same FL PHY frame and these x-FLABs do not have the same values in all fields except the NodeID, SZ, and Supplemental fields, the access terminal shall determine that an error has occurred. In this case, the access terminal shall ignore all these x-FLABs. Additionally, the remaining steps shown in this subclause shall not be carried out.
- e) If all the received x-FLABs have a common packet format field of NULL, the access terminal shall carry out the following operations:
 - If all the received x-FLABs do not have a common value for the Supplemental field, then the access terminal shall determine that an error has occurred. In this case, the access terminal shall ignore all these x-FLABs.
 - If all the received x-FLABs have the Supplemental field set to '0', the access terminal shall treat this set of x-FLABs as a single x-FLAB for Resource Reassignment. The final set of hop-ports after the Resource Reassignment operation shall be given by $f_{\text{ATA\_Computation}}(\Phi, N_0, N_1, \dots)$, where the function $f_{\text{ATA\_Computation}}$ is as defined in 8.6.5.5.1.1.1.1 and Φ denotes the empty set of hop-ports. In the above, a set of FLABs causes Resource Reassignment of an FL-ATA, a set of NP-FLAB_{UCS} causes Resource Reassignment of an FL-NP-ATA_{UC}, a set of NP-FLAB_{BCS} causes Resource Reassignment of an FL-NP-ATA_{BC} and a set of RRA-FLABs causes Resource Reassignment of an FL-R-ATA. The Resource Reassignment operation for FL-ATAs, FL-NP-ATA_s (unicast and broadcast) and FL-R-ATAs is described in 8.6.5.5.1.1.3.3, 8.6.5.5.1.1.4.3, and 8.6.5.5.1.1.5.3 respectively. If the received x-FLABs are of type MCW, then any x-MCW-FLAB2 received in the same FL PHY frame shall be ignored.
 - If all the received x-FLABs have the Supplemental field set to '1', the access terminal shall treat this set of x-FLABs as a single x-FLAB for HARQ extension. The final set of hop-ports after the HARQ extension operation shall be given by N_{init} if $N_0 = \Phi$, and by $f_{\text{ATA\_Computation}}(\Phi, N_0, N_1, \dots)$ otherwise. Here, the function $f_{\text{ATA\_Computation}}$ is as defined in

8.6.5.5.1.1.1.1, Φ denotes the empty set of hop-ports, and N_{init} denotes the set of hop-ports in the current x-ATA. In the above, a set of FLABs causes HARQ extension of an FL-ATA, a set of NP-FLAB_{UC}s causes HARQ extension of an FL-NP-ATA_{UC}, a set of NP-FLAB_{BC}s causes HARQ extension of an FL-NP-ATA_{BC} and a set of RRA-FLABs causes HARQ extension of an FL-R-ATA. The HARQ extension operation for FL-ATAs, FL-NP-ATA_{UC}, FL-NP-ATA_{BC}, and FL-R-ATA is described in 8.6.5.5.1.1.3.4, 8.6.5.5.1.1.4.4, and 8.6.5.5.1.1.5.4 respectively. If the received x-FLABs are of type MCW, then any x-MCW-FLAB2 received in the same FL PHY frame shall be ignored.

- f) If all the received x-FLABs have a common packet format field that is not equal to NULL, the access terminal shall carry out the following operations:
- If any x-FLAB except the one with the lowest index pair has the Supplemental field set to '0', the access terminal shall determine that an error has occurred. In this case, the access terminal shall ignore all these x-FLABs.
 - If the x-FLAB with the lowest index pair has its supplemental field set to '0' and all the remaining x-FLABs have their supplemental field set to '1', the access terminal shall treat this as a new assignment (non-supplemental assignment in the case of FL-ATAs). Additionally, if the received x-FLABs are of type MCW, and no x-MCW-FLAB2 is received in the same FL PHY frame, the access terminal shall determine that an error has occurred and shall ignore all these x-FLABs. The set of hop-ports assigned by this non-supplemental assignment shall be given by $f_{ATA\_Computation}(\Phi, N_0, N_1, \dots)$, where the function $f_{ATA\_Computation}$ is as defined in 8.6.5.5.1.1.1.1 and Φ denotes the empty set of hop-ports. New assignments are described for FL-ATAs, FL-NP-ATAs (unicast and broadcast), and FL-R-ATAs in 8.6.5.5.1.1.3.2, 8.6.5.5.1.1.4.2, and 8.6.5.5.1.1.5.1 respectively.
 - If all the received x-FLABs have their supplemental field set to '1', the access terminal shall treat this as a supplemental assignment for the case of FLABs and NP-FLAB_{UC}s. For the case of NP-FLAB_{BC}s and RRA-FLABs, the access terminal shall determine that an error has occurred and shall ignore all these x-FLABs. Supplemental assignments modify an existing FL-ATA and are described in 8.5.5.4.1.1.3.2. The final set of hop-ports after the application of this supplemental assignment shall be given by $f_{ATA\_Computation}(N_{init}, N_0, N_1, \dots)$ where N_{init} denotes the set of hop-ports in the existing FL-ATA and the function $f_{ATA\_Computation}$ is as defined in 8.6.5.5.1.1.1.1.

8.6.5.5.1.1.1.1 Definition of $f_{ATA\_Computation}$

The function $f_{ATA\_Computation}$ takes a set of at least two input arguments in order, each of which denotes a set of hop-ports and provides an output that is another set of hop-ports. The input set of hop-ports may be represented by a NodeID or SZ field of an F-SCCH block.

The output of the function $f_{ATA\_Computation}$ is computed according to the following procedure:

- a) Let n denote the number of input arguments to the function $f_{ATA\_Computation}$. Label the different arguments $T_0, T_1, \dots, T_{n-1}$. Initialize a counter i to 1, and a hop-port set T_{out} to T_0 .
- b) If T_i is a subset of T_{out} , then set T_{out} to be the set of hop-ports that are contained in T_{out} but not in T_i ; else set T_{out} to be the union of T_i and T_{out} .
- c) Increment i . If $i < n$, go to step b).

The output of the function $f_{ATA\_Computation}$ is given by the hop-port set T_{out} at the end of the above procedure.

8.6.5.5.1.1.2 Access terminal rules for overlapping assignments

The following rules shall be followed in order to satisfy the constraints specified in 8.6.5.3.2. Here, a new ATA refers to an ATA that has been created in this FL PHY frame, and excludes modifications to an existing ATA through supplemental assignments, Resource Reassignment, and HARQ extension operations.

- a) If a new FL-ATA/FL-NP-ATA_{UC}/FL-R-ATA overlaps in time with a previously existing instance of any one of an FL-ATA, an FL-NP-ATA_{UC}, or an FL-R-ATA, the new ATA takes precedence and the already existing ATA is expired. If an FL-ATAs/FL-NP-ATA_{UC}s/FL-R-ATAs is created in the same FL PHY frame as any one of an FL-ATA, FL-NP-ATA_{UC}, or an FL-R-ATA, then both ATAs shall be expired.
- b) If a new FL-NP-ATA_{BC} overlaps in time with a previously existing FL-NP-ATA_{BC}, the new ATA takes precedence and the already existing ATA is expired.
- c) If a new FL-NP-ATA_{UC}/FL-R-ATA overlaps in both time and frequency with a previously existing FL-NP-ATA_{BC} or vice-versa, the new ATA takes precedence and the already existing ATA is expired. If an FL-NP-ATA_{UC}/FL-R-ATA is created in the same FL PHY frame as an FL-NP-ATA_{BC}, and the two overlap in frequency, then both ATAs shall be expired.
- d) If a new FL-ATA overlaps in both time and frequency with a previously existing FL-NP-ATA_{BC}, then the new ATA takes precedence and the already existing ATA shall be expired.
- e) If a new FL-NP-ATA_{BC} overlaps in both time and frequency with a previously existing FL-ATA or concurrently created FL-ATA, then the FL-ATA shall be decremented by the set overlapping hop-ports in frequency.

8.6.5.5.1.1.3 Access terminal assignment management for persistent assignments

The logic in this subclause assumes that all of the FLABs/NP-FLABs are being sent from the same serving sector, the FLSS. The logic for access terminal assignment management during handoff is found in 8.6.5.5.1.1.7.

Upon receiving an *AirLinkManagement.TunedAway* indication, the access terminal shall expire all its FL-ATAs.

The access terminal shall perform the following steps in each FL PHY frame to update its FL-ATAs:

- a) Follow the procedure described in 8.6.5.5.1.1.1 for updating the FL-ATA, FL-NP-ATA_{UC}, FL-NP-ATA_{BC}, and FL-R-ATA based on the received assignment blocks.
- b) After FL-ATA, FL-NP-ATA_{UC}, FL-NP-ATA_{BC}, and FL-R-ATA have all been updated, follow the procedures described in 8.6.5.5.1.1.2 for handling overlapping assignments.
- c) Follow the procedure specified in 8.6.5.5.1.1.3.5 for acknowledging assignments.

8.6.5.5.1.1.3.1 Deassigning assignments

If a received FLAB or F-SPCH indication causes deassignment of the FL-ATA, as determined in 8.6.5.5.1.1.1, then the FL-ATA for the given set of PHY frames shall be expired.

8.6.5.5.1.1.3.2 Supplemental and non-supplemental assignments

8.6.5.5.1.1.3.2.1 Supplemental assignments

An existing FL-ATA can have its set of hop-ports modified via supplemental_assignments, as determined in 8.6.5.5.1.1.1. The supplemental assignment is persistent or non-persistent, depending on whether it is caused by FLABs or NP-FLAB_{UC}s, respectively. Supplemental assignments can also be used to change other fields of an existing FL-ATA such as packet format, without changing the resources.

If a persistent supplemental assignment is received from a particular set of PHY frames, then the following are true:

- a) If the FL-ATA for the given set of PHY frames is empty, or the FL-ATA for the given set of PHY frames has a type that does not match the received FLABs, or a different SubtreeID than the received FLABs, the access terminal shall determine that an error condition has occurred and shall ignore the received FLABs.
- b) If the FL-ATA for the given set of PHY frames is not empty, and the FL-ATA has the same type as the received FLABs, and the SubtreeID is the same as the received FLABs, then the following are true:
 - 1) The hop-ports of the FL-ATA shall be modified as described in 8.6.5.5.1.1.1.
 - 2) The new PF shall be used in place of any PFs that may have been specified in any previous assignment. Similarly, the FL Pilot, TD, and NumLayers shall be updated according to the new FLAB (when relevant for the particular kind of FLAB). If the FL-ATA is of type MCW, and if an MCW-FLAB2 block is received in the same FL PHY frame, then the packet formats for all layers other than layer 1 shall be given by the PF fields in the MCW-FLAB2 block. Otherwise, the packet formats for the existing FL-ATA shall be retained for these layers.

If a non-persistent supplemental assignment is received for a particular set of PHY frames, then the following are true:

- a) If the FL-ATA for the given set of PHY frames is empty, or the FL-ATA for the given set of PHY frames has a type that does not match the received NP-FLAB_{UCS}, or the SubtreeID is different from the received NP-FLAB_{UCS}, the access terminal shall determine that an error condition has occurred and shall ignore the NP-FLAB_{UCS}.
- b) If the FL-ATA for the given set of PHY frames is not empty, and the FL-ATA has the same type as the received NP-FLAB_{UCS}, and the SubtreeID is the same as the one in the received NP-FLAB_{UCS}, then the following changes shall be made for the duration of only one packet, and the original FL-ATA shall take effect when the packet is decoded by the Physical Layer.
 - 1) The hop-ports of the FL-ATA shall be modified as described in 8.6.5.5.1.1.1.
 - 2) The new PF shall be used in place of any PFs that may have been specified in any previous assignment. Similarly, the FL Pilot, TD, and NumLayers will be updated according to the new NP-FLAB_{UCS} (when relevant for the particular kind of FLAB). If the FL-ATA is of type MCW, and if an MCW-FLAB2 block is received in the same FL PHY frame, then the packet formats for all layers other than layer 1 shall be given by the PF fields in the MCW-FLAB2 block. Otherwise, the packet formats for the existing FL-ATA shall be retained for these layers.

8.6.5.5.1.1.3.2.2 Non-supplemental assignments

If a non-supplemental assignment is received, as determined in 8.6.5.5.1.1.1, then the FL-ATA for the set of PHY frames assigned by the received FLABs shall be cleared before adding the hop-ports specified by the received FLABs, as determined in 8.6.5.5.1.1.1, to the ATA for the given set of PHY frames.

The PF, FL Pilot, TD, and NumLayers fields in the received FLABs shall be used for the new FL-ATA (when relevant for the particular type of FLAB). If the FL-ATA is of type MCW, then the packet formats for all layers other than layer 1 are given by the PF fields in the received MCW-FLAB2 block.

8.6.5.5.1.1.3.3 Resource reassignment

If a resource reassignment is received, as determined in 8.6.5.5.1.1.1, then the following are true:

- a) If the FL-ATA for the given set of PHY frames is empty, or the FL-ATA for the given set of PHY frames has a type that does not match the received FLABs, or if the FL Pilot, TD, NumLayers, or SubTreeID for the FL-ATA is different than the ones in the received FLABs (when applicable), the

access terminal shall determine that an error condition has occurred and shall ignore the received FLABs.

- b) If the FL-ATA for the given set of PHY frames is not empty, and the FL-ATA has the same type as the received FLABs, and the FL Pilot, TD, NumLayers, and SubTreeID for the FL-ATA is the same for the received FLABs (when applicable), then the FL-ATA shall be changed to the hop-ports determined by 8.6.5.5.1.1.1 for the HARQ transmission beginning in the same FL PHY frame. The previous PF is maintained.

8.6.5.5.1.1.3.4 HARQ extension

A persistent assignment can have the total number of HARQ transmissions increased for one packet via HARQ extension.

If a HARQ extension is received for a particular set of PHY frames, as determined in 8.6.5.5.1.1.1, then the following are true:

- a) If the FL-ATA for the given set of PHY frames is empty, or has a type that does not match the received FLABs, or if the FL Pilot, TD, NumLayers, or SubTreeID for the FL-ATA is different than the ones in the received FLABs (when applicable), the access terminal shall determine that an error condition has occurred and shall ignore the received FLABs.
- b) If the FL-ATA for the given set of PHY frames is not empty, and the FL-ATA has the same type as the received FLABs, and the FL Pilot, TD, NumLayers, and SubTreeID for the FL-ATA is the same for the received FLABs (when applicable), then there will be six additional HARQ transmissions for the packet currently being decoded on the FL-ATA, starting with the FL PHY frame in which the HARQ extension was received. Additionally, the set of hop-ports shall be moved to the set of hop-ports determined in 8.6.5.5.1.1.1. The previous PF is maintained.

HARQ transmissions of a given packet can be extended multiple times, using HARQ extension assignment blocks.

8.6.5.5.1.1.3.5 Assignment acknowledgment

The access terminal shall issue a ReverseControlChannelMAC.ACKAssignment along with the index k if the following conditions are satisfied (note that multiple such commands may be issued in each FL PHY frame):

- a) The access terminal received an FLAB with index pair $(k,0)$, where the index pair is as defined in 8.5.5.4.1.1.3.
- b) The value of AssignmentAckHARQTransmission is not 0.
- c) There have been AssignmentAckHARQTransmission HARQ transmissions on the changed FL-ATA.
- d) The F-DCH packet being transmitted on the FL-ATA has not been successfully decoded.

Assignment acknowledgements follow the same timing structure as ACKs (see 8.1.5.1).

8.6.5.5.1.1.4 Access terminal assignment management for non-persistent assignments

The logic in this subclause assumes that all of the FLABs/NP-FLABs are being sent from the FLSS. Upon receiving an *AirLinkManagement.TunedAway* indication, the access terminal shall expire all its FL-NP-ATAs.

The access terminal shall give up a FL-NP-ATA_{BC} after it has kept it for the maximum number of six HARQ transmissions, unless it receives assignment blocks for HARQ extension, as determined in 8.6.5.5.1.1.1.

Similarly, the access terminal shall give up a FL-NP-ATA_{UC} after it has kept it the maximum number of six HARQ transmissions, unless it receives assignment blocks for HARQ extension, as determined in 8.6.5.5.1.1.1.

The access terminal shall perform the following steps in each FL PHY frame to update its FL-NP-ATAs:

- a) Follow the procedure described in 8.6.5.5.1.1.1 for updating the FL-ATA, FL-NP-ATA_{UC}, FL-NP-ATA_{BC}, and FL-R-ATA based on the received assignment blocks.
- b) After FL-ATA, FL-NP-ATA_{UC}, FL-NP-ATA_{BC}, and FL-R-ATA have all been updated, follow the procedures described in 8.6.5.5.1.1.2 for handling overlapping assignments.
- c) Follow the procedure specified in 8.6.5.5.1.1.4.5 for acknowledging assignments.

8.6.5.5.1.1.4.1 Deassigning assignments

If a received NP-FLAB_{UC} or F-SPCH indication causes deassignment of the FL-NP-ATA_{UC}, as determined in 8.6.5.5.1.1.1, then any FL-NP-ATA_{UC} for the given set of PHY frames shall be expired.

If a received NP-FLAB_{BC} causes deassignment of the FL-NP-ATA_{BC}, as determined in 8.6.5.5.1.1.1, then any FL-NP-ATA_{BC} for the given set of PHY frames shall be expired.

8.6.5.5.1.1.4.2 New non-persistent assignments

If a new non-persistent unicast assignment is received for a given set of PHY frames, as determined in 8.6.5.5.1.1.1, any previous FL-NP-ATA_{UC} on the given set of PHY frames shall be expired and the new FL-NP-ATA_{UC} shall consist of the set of hop-ports determined in 8.6.5.5.1.1.1.

The PF, FL Pilot, TD, and NumLayers fields in the received NP-FLAB_{UCS} shall be used for the new FL-NP-ATA_{UC} (when relevant for the particular type of FLAB). If the FL-NP-ATA_{UC} is of type MCW, then the packet formats for all layers other than layer 1 are given by the PF fields in the received MCW-NP-FLAB2 block.

If a new non-persistent broadcast assignment is received for a given set of PHY frames, as determined in 8.6.5.5.1.1.1, any previous FL-NP-ATA_{BC} on the given set of PHY frames shall be expired, and the new FL-NP-ATA_{BC} shall consist of the set of hop-ports determined in 8.6.5.5.1.1.1.

8.6.5.5.1.1.4.3 Resource reassignment

If a unicast resource reassignment is received, as determined in 8.6.5.5.1.1.1, then the following are true:

- 1) If the FL-NP-ATA_{UC} for the given set of PHY frames is empty, or has a type that does not match the received NP-FLAB_{UCS}, or the FL Pilot, TD, NumLayers, and SubTreeID for the FL-NP-ATA_{UC} are different than the ones in the received NP-FLAB_{UCS} (when applicable), the access terminal shall determine that an error condition has occurred and shall ignore the received NP-FLAB_{UCS}.
- 2) If the FL-NP-ATA_{UC} for the given set of PHY frames is not empty, and has the same type as the received NP-FLAB_{UCS}, and the FL Pilot, TD, NumLayers, and SubTreeID for the FL-NP-ATA_{UC} are the same as the ones in the received NP-FLAB_{UCS} (when applicable), then the FL-NP-ATA_{UC} shall be changed to the set of hop-ports determined in 8.6.5.5.1.1.1 for the HARQ transmission beginning in the same FL PHY frame. The previous PF is maintained.

If a broadcast resource reassignment is received, as determined in 8.6.5.5.1.1.1, then the following are true:

- 1) If the FL-NP-ATA_{BC} for the given set of PHY frames is empty, the access terminal shall determine that an error condition has occurred and shall ignore the received NP-FLAB_{BCs}.
- 2) If the FL-NP-ATA_{BC} for the given set of PHY frames is not empty, then the FL-NP-ATA_{BC} shall be changed to the set of hop-ports determined in 8.6.5.5.1.1.1 for the HARQ retransmission beginning in the same FL PHY frame. The previous PF is maintained.

8.6.5.5.1.1.4.4 HARQ extension

A non-persistent assignment can have the total number of HARQ transmissions increased via HARQ extension.

If a unicast HARQ extension is received for a particular set of PHY frames, as determined in 8.6.5.5.1.1.1, then the following are true:

- a) If the FL-NP-ATA_{UC} for the given set of PHY frames is empty, or has a type that does not match the received NP-FLAB_{UCs}, or if the FL Pilot, TD, NumLayers, and SubTreeID for the FL-NP-ATA_{UC} are different than the ones in the received NP-FLAB_{UCs} (when applicable), the access terminal shall determine that an error condition has occurred and shall ignore the received NP-FLAB_{UCs}.
- b) If the FL-NP-ATA_{UC} for the given set of PHY frames is not empty, and the FL-NP-ATA_{UC} has the same type as the received NP-FLAB_{UCs}, and the FL Pilot, TD, NumLayers, and SubTreeID for the FL-NP-ATA_{UC} are the same as the ones in the received NP-FLAB_{UCs} (when applicable), then there will be six additional HARQ transmissions for the packet currently being decoded on the FL-NP-ATA_{UC}, starting with the FL PHY frame in which the HARQ extension was received. Additionally, the set of hop-ports shall be moved to the set of hop-ports determined in 8.6.5.5.1.1.1.

If a broadcast HARQ extension is received for a particular set of PHY frames as determined in 8.6.5.5.1.1.1, then the following are true:

- If the FL-NP-ATA_{BC} for the given set of PHY frames is empty, the access terminal shall determine that an error condition has occurred and shall ignore the received NP-FLAB_{BCs}.
- If the FL-NP-ATA_{BC} for the given set of PHY frames is not empty, then there will be six additional HARQ transmissions for the packet currently being decoded on the FL-NP-ATA_{BC}, starting with the FL PHY frame in which the HARQ extension was received. Additionally, the set of hop-ports shall be moved to the set of hop-ports determined in 8.6.5.5.1.1.1. The previous PF is maintained.

HARQ transmissions of a given packet can be extended multiple times, using HARQ extension assignment blocks.

8.6.5.5.1.1.4.5 Assignment acknowledgement

The access terminal shall issue a *ReverseControlChannelMAC.ACKAssignment* along with the index k if the following conditions are satisfied (note that multiple such commands may be issued in each FL PHY frame):

- a) The access terminal received an NP-FLAB_{UC} with index pair (k,0), where the index pair is as defined in 8.5.5.4.1.1.3.6.
- b) The value of AssignmentAckHARQTransmission is not 0.
- c) There have been AssignmentAckHARQTransmission HARQ transmissions on the changed FL-NP-ATA_{UC}.
- d) The F-DCH packet being transmitted on the FL-NP-ATA_{UC} has not been successfully decoded.

Assignment acknowledgements follow the same timing structure as ACKs (see 8.1.5.1).

8.6.5.5.1.1.5 Access terminal assignment management for residual resource assignments

Upon receiving an *AirLinkManagement.TunedAway* indication, the access terminal shall expire all its FL-R-ATAs.

The logic in this subclause assumes that all of the RRA-FLABs are being sent from the FLSS.

The access terminal shall give up a FL-R-ATA after it has kept it for the maximum number of six HARQ transmissions, unless it receives assignment blocks for HARQ extension, as determined in 8.6.5.5.1.1.1. The access terminal shall also give up a FL-R-ATA if the packet decodes successfully before the maximum number of HARQ transmissions is reached.

The access terminal shall perform the following steps for each FL PHY frame to update its FL-R-ATAs:

- a) Follow the procedure described in 8.6.5.5.1.1.1 for updating the FL-ATA, FL-NP-ATA_{UC}, FL-NP-ATA_{BC}, and FL-R-ATA based on the received assignment blocks.
- b) After FL-ATA, FL-NP-ATA_{UC}, FL-NP-ATA_{BC}, and FL-R-ATA have all been updated, follow the procedures described in 8.6.5.5.1.1.2 for handling overlapping assignments.
- c) In each FL PHY frame, follow the procedure described in 8.6.5.5.1.1.4 to determine the FL-R-ATA for that FL PHY frame.
- d) Follow the procedure specified in 8.6.5.5.1.1.5.6 for acknowledging assignments.

8.6.5.5.1.1.5.1 New residual resource assignments

If a new residual resource assignment is received for a given set of PHY frames, as determined in 8.6.5.5.1.1.1, any previous FL-R-ATA_{UC} on the given set of PHY frames shall be expired, and the new FL-R-ATA shall consist of the set of hop-ports determined in 8.6.5.5.1.1.1.

The PF, FL Pilot, TD, and NumLayers fields in the received RRA-FLABs shall be used for the new FL-R-ATA (when relevant for the particular type of RRA-FLAB). If the FL-R-ATA is of type MCW, then the packet formats for all layers other than layer 1 are given by the PF fields in the received MCW-RRA-FLAB2 block.

8.6.5.5.1.1.5.2 Deassigning assignments

If a received RRA-FLAB causes deassignment of the of the FL-R-ATA, as determined in 8.6.5.5.1.1.1, then any FL-R-ATA for the given set of PHY frames shall be expired.

8.6.5.5.1.1.5.3 Resource reassignment

If a resource reassignment is received, as determined in 8.6.5.5.1.1.1, then the following are true:

- a) If the FL-R-ATA for the given set of PHY frames is empty, or the FL-R-ATA for the given set of PHY frames has a type that does not match the received RRA-FLABs, or if the FL Pilot, TD, NumLayers, or SubTreeID for the FL-ATA is different than the ones in the received RRA-FLABs (when applicable), the access terminal shall determine that an error condition has occurred and shall ignore the received RRA-FLABs.
- b) If the FL-R-ATA for the given set of PHY frames is not empty, and the FL-R-ATA has the same type as the received RRA-FLABs, and the FL Pilot, TD, NumLayers, and SubTreeID for the FL-R-ATA is the same as for the received FLABs (when applicable) then the FL-R-ATA shall be changed to the set of hop-ports determined in 8.6.5.5.1.1.1 for the HARQ retransmission beginning in the same FL PHY frame. The previous PF is maintained.

8.6.5.5.1.1.5.4 HARQ extension

A FL-R-ATA can have the total number of HARQ transmissions increased via HARQ extension.

- a) If a HARQ extension is received for a particular set of PHY frames, as determined in 8.6.5.5.1.1.1, then the following are true:
- b) If the FL-R-ATA for the given set of PHY frames is empty, or has a type that does not match the received RRA-FLABs, or if the FL Pilot, TD, NumLayers, and SubTreeID for the FL-R-ATA are different than the ones in the received RRA-FLABs (when applicable), then the access terminal shall determine that an error condition has occurred and shall ignore all the received RRA-FLABs.
- c) If the FL-R-ATA for the given set of PHY frames is not empty, and the FL-R-ATA has the same type as the received RRA-FLABs, and the FL Pilot, TD, NumLayers, and SubTreeID for the FL-R-ATA are the same as the ones in the received RRA-FLABs (when applicable), then there will be six additional HARQ transmissions for the packet currently being decoded on the FL-R-ATA, starting with the FL PHY frame in which the HARQ extension was received. Additionally, the set of hop-ports shall be moved to the set of hop-ports determined in 8.6.5.5.1.1.1.

HARQ transmissions of a given packet can be extended multiple times, using HARQ extension assignment blocks.

8.6.5.5.1.1.5.5 Determining the final FL-R-ATA

For each FL-R-ATA assigned by an RRA-FLAB, the access terminal shall start with the hop-ports assigned by the last received set of RRA-FLABs, and remove all hop-ports as indicated by the GRA bitmap and any hop-ports used for transmitting the GRA bitmap as described in 8.6.5.5.1.1.6.4.8. If no GRA bitmap is received along with the RRA-FLAB, then the access terminal shall expire all FL-R-ATAs on the corresponding set of FL PHY frames. Every time a new GRA bitmap is received, the FL-R-ATA shall be recomputed, until the FL-R-ATA expires.

8.6.5.5.1.1.5.6 Assignment acknowledgment

The access terminal shall issue a *ReverseControlChannelMAC.ACKAssignment* along with the index k if the following conditions are satisfied (note that multiple such commands may be issued in each FL PHY frame):

- a) The access terminal received an RRA-FLAB with index pair $(k,0)$, where the index pair is as defined in 8.5.5.4.1.1.3.6, and has also received a GRA bitmap in the same FL PHY frame.
- b) The value of AssignmentAckHARQtransmission is not 0.
- c) There have been AssignmentAckHARQtransmission HARQ transmissions on the new/changed FL-R-ATA.
- d) The F-DCH packet being transmitted on the FL-R-ATA has not been successfully decoded.

Assignment acknowledgements follow the same timing structure as ACKs (see 8.1.5.1).

8.6.5.5.1.1.6 Access terminal assignment management for Group Resource Allocations

Group Resource Allocation (GRA) is a scheme whereby an access terminal determines its resource assignment in a dynamic manner based on one or more GRA bitmaps received from the access network and knowledge of a set of shared resources (group resources).

8.6.5.5.1.1.6.1 Group Index

The access network establishes groups of access terminals that monitor one or more shared GRA bitmaps to determine their respective resource allocation. Each group of access terminals is identified by a Group

Index. The Group Index is transmitted from the access network to the access terminal using the GroupIndex field of the Group Assignment Message (GAM).

If the access network indicates an access terminal shall monitor other groups in the GAM, the resources assigned to other groups (RAOG) shall be determined in order to determine available resources. The access terminal shall determine its resource allocation from the GRA bitmap associated with its Group Index and the GRA bitmaps associated with OtherGroupIndex.

8.6.5.5.1.1.6.2 Group resources

The access network shall assign each group a set of shared time-frequency resources, denoted as *group resources*. In the time domain, the group resources can be one the following:

- FL 8-interlace structure with two PHY frame transmission in the FDD mode; not applicable to the TDD mode.
- FL 8-interlace structure without extended transmission in the FDD mode; FL 4-interlace structure in the TDD mode.
- FL 6-interlace structure with two PHY frame transmission in the FDD mode; not applicable to the TDD mode.
- FL 6-interlace structure without extended transmission in the FDD mode; FL three-interlace structure in the TDD mode.

The term *GRA frame* will be used to denote the time domain resource. In the frequency domain, the group resources correspond to nodes from subtree 0 of the channel tree.

If the GRA frame is defined as one PHY frame, the group resources can be numbered, in terms of base-nodes, using one of the following two ordering schemes. First, the base nodes of the group resources are numbered beginning with the base-node with the smallest index, followed by the base-node with the second smallest index, and so on. Second, the base nodes of the group resources are numbered beginning with the base-node with the largest index, followed by the base-node with the second largest index, and so on.

If the GRA frame is defined as two frames, the group resources can be numbered, in terms of PHY frames and base-nodes, using one of the following two ordering schemes. First, the group resources can be numbered beginning with the base-node with the smallest index, in the first PHY frame, followed by the base-node with the smallest index in the second PHY frame, followed by the base-node with the second smallest index in the first PHY frame, followed by the base-node with the second smallest index in the second PHY frame, and so on. Second, the group resources can be numbered beginning with the base-node with the largest index in the second PHY frame, followed by the base-node with the largest index in the first PHY frame, followed by the base-node with the second largest index in the second PHY frame, followed by the base-node with the second largest index in the first PHY frame, and so on.

An indication of the group resources and the corresponding ordering scheme are transmitted from the access network to the access terminal using the GAM. The term *resource* will be used to represent a series of N sequential base-nodes (sequential in time and frequency according to the ordering scheme defined above) within the group resources, where an indication of N is transmitted from the access network to the access terminal using the ResourceSize field of the GAM.

8.6.5.5.1.1.6.3 Group interlace

The access network shall assign each group to one or more group interlaces, where a group interlace is a repeating sequence of GRA frames. The group with GroupIndex set to '000001' shall be defined over all interlaces, and the remaining groups shall be assigned to a group interlace using the GroupInterlace field of the GAM. The group interlace is defined according to Equation (2).

$$\text{GROUP\_INTERLACE} = \left\lfloor \frac{\text{FRAME\_INDEX} + \text{SUPERFRAME\_INDEX} \times N_{\text{FLPHYFrames}}}{N_{\text{GROUPFRAMES}}} \right\rfloor \bmod N_{\text{GROUPINTERLACES}}$$

(2)

where

- FRAME_INDEX
- is the index of the FL PHY frame within the superframe
- SUPERFRAME_INDEX
- is the index of the current superframe
- N_{FLPHYFrames}
- is the number of FL PHY frames that make up a superframe
- N_{GROUPFRAMES}
- is the number of FL PHY frames that make up the GRA frame
- N_{GROUPINTERLACES}
- is the total number of group interlaces

8.6.5.5.1.1.6.4 Group Resource Allocation bitmap

The structure of the GRA bitmap is shown in Figure 57.

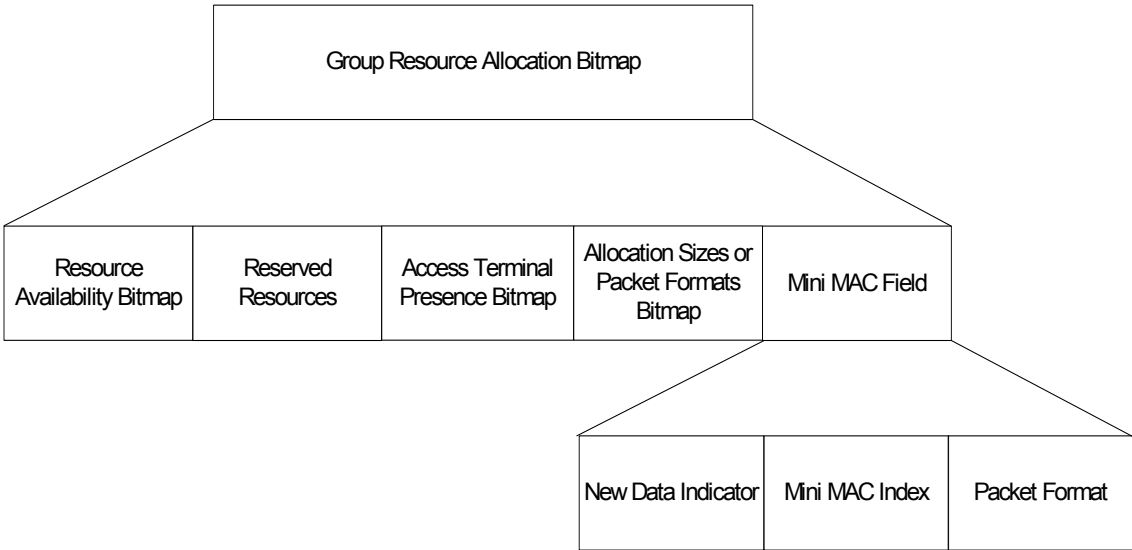


Figure 57—GRA bitmap structure

Referring to Figure 57, the GRA bitmap is comprised of a subset of the following—a resource availability bitmap, a reserved resources field, an access terminal presence bitmap, an allocation sizes or packet formats bitmap, and a mini MAC field. The mini MAC field is comprised of a subset of the following—a new data indicator field, a mini MAC index field, and a packet format field.

8.6.5.5.1.1.6.4.1 Resource availability bitmap

The resource availability bitmap may be omitted from the group resource allocation bitmap, except for the group whose GroupIndex is set to '000001'. If used, each bit in the resource availability bitmap corresponds to one resource. The access network shall set a bit in the resource availability bitmap to '1' to indicate that the corresponding resource is currently being used and shall set a bit in the resource availability to '0' to indicate that the corresponding resource is available. The resource availability bitmap may have fewer bits than are necessary to represent each resource within the group resources, in which case the bits in the resource availability bitmap correspond to the lowest numbered resources.

8.6.5.5.1.1.6.4.2 Reserved resources field

The reserved resources field may be omitted from the group resource allocation bitmap. If used, the access network shall set the reserved resources field to the number of resources, beginning with the first available resource, that are being used by other access terminals.

8.6.5.5.1.1.6.4.3 Access terminal presence bitmap

The access network shall transmit the access terminal presence bitmap as part of the group resource allocation bitmap, except for the group whose GroupIndex is set to '000001'. Each bit in the access terminal presence bitmap corresponds to one or more access terminals. The access network shall set a bit in the access terminal presence bitmap to '1' to indicate that the corresponding access terminal(s) are active; i.e., the access network is transmitting a packet to the access terminal(s) using the resource(s) derived from the position of the bit within the access terminal presence bitmap and shall set a bit in the access terminal presence bitmap to '0' to indicate that the corresponding access terminal(s) are not active; i.e., the access network is not transmitting a packet to the access terminal(s) using the resource(s) derived using the position of the bit within the access terminal presence bitmap.

The access network can configure the resource availability bitmap and access terminal presence bitmap of a group such that the two bitmaps have different lengths in the different group interlace offsets. The length of the resource availability bitmap plus the length of the access terminal presence bitmap shall be the same for all group interlace offsets.

8.6.5.5.1.1.6.4.4 Allocation sizes or packet formats bitmap

The allocation sizes or packet formats bitmap may be omitted from the group resource allocation bitmap. If used, the allocation sizes or packet formats bitmap is used, in conjunction with the access terminal presence bitmap, to indicate the number of resources allocated to each active access terminal or the packet format assigned to each active access terminal. The allocation sizes or packet formats bitmap contains B bits per active access terminal, where an indication of B is transmitted from the access network to the access terminal using the AllocationSizesOrPacketFormatsBitsPerActiveAT field of the GAM. The L^{th} set of B bits in the allocation sizes or packet formats bitmap corresponds to the L^{th} active access terminal as indicated in the access terminal presence bitmap. The allocation sizes or packet formats bitmap can be interpreted in one of the following three ways:

- a) An allocation sizes bitmap is used. The number of allocated resources is one more than the binary value indicated in the allocation sizes bitmap for the access terminal. A single packet is encoded and transmitted on the number of allocated resources.
- b) An allocation sizes bitmap is used. The number of allocated resources is one more than the binary value indicated in the allocation sizes bitmap for the access terminal.
 - 1) If one resource is allocated, a single packet is encoded and transmitted on the resource.
 - 2) If more than one resource is allocated,
 - Two individually encoded packets (P_1 and P_2) are transmitted on evenly divided portions of NAR, where NAR is the number of allocated resources. P_2 is transmitted on the first $\lfloor \text{NAR}/2 \rfloor$ resources within the access terminal's allocation, and P_1 is transmitted on the last $\lceil \text{NAR}/2 \rceil$ resources within the access terminal's allocation.
 - The HARQ transmission opportunity for P_2 is less than the HARQ transmission opportunity for P_1 by the number of group interlace offsets, where an indication of the number of group interlace offsets is transmitted from the access network to the access terminal using the NumGroupInterlaceOffsets field of the GAM.
- c) The packet formats bitmap is used. The packet format used for the access terminal is indicated in the packet formats bitmap. The value in the packet formats bitmap corresponds to one of the entries in the list of GroupPacketFormats as defined by the GAM.

The access network shall transmit an indication of how the allocation sizes or packet formats bitmap should be interpreted using the AllocationSizesOrPacketFormatsBitmapInterpretation field of the GAM.

8.6.5.5.1.1.6.4.5 Mini MAC field

The mini MAC field may be omitted from the group resource allocation bitmap. If the mini MAC field is used, and there are a total of T available resources within the group resources, then the mini MAC field is used to allocate the K^{th} to $T-1^{\text{th}}$ available resources, subject to a maximum number of resources, to one access terminal. K is defined according to Equation (3).

$$K = RR + NR \quad (3)$$

where

- RR is the number of resources that are indicated as being used by the reserved resources field (if present)
- NR is the number of resources that are allocated to access terminals with positions that are within the access terminal presence bitmap

The set of available resources (AR) is defined according to Equation (4).

$$AR = (R - RABR) - RAOG \quad (4)$$

where

- R is the list of group resources
- $RABR$ is the list of those resources that are indicated as being used in the resource availability bitmap
- $RAOG$ is the list of resources that have been allocated to other groups

An indication of the maximum number of resources is transmitted from the access network to the access terminal using the MiniMACMaxResources field of the GAM. If $T - K$ is greater than the value indicated in the MiniMACMaxResources field, the access network shall use the K^{th} to $K + \text{MiniMACMaxResources} - 1$ available resources for transmitting the packet to the access terminal indicated in the mini MAC field.

The mini MAC field is divided into three parts. First, the mini MAC field shall consist of a new data indicator (NDI) field. Second, the mini MAC field shall consist of the mini MAC index of the access terminal for which the packet is intended. Third, the mini MAC field may contain an indication of the packet format of the packet. The access network shall set the NDI field to '1' when it is transmitting the first HARQ subpacket for an access terminal; otherwise, the NDI field shall be set to '0'. The access network shall set the mini MAC index portion of the mini MAC field to the mini MAC index of the targeted access terminal, where a mini MAC index is assigned to the access terminal by the access network using the MiniMACIndex field of the GAM.

If a packet format field is included as part of the mini MAC field, the access network shall set the packet format field to the value corresponding to the packet format for the targeted access terminal.

If the GRA bitmap consists of an access terminal presence bitmap, an allocation sizes or packet formats bitmap, and a mini MAC field, then the last R_{MM} entries of the allocation sizes or packet formats bitmap shall be interpreted as a special resource availability bitmap if R_{MM} is greater than 0, where R_{MM} is defined as shown in Equation (5).

$$R_{\text{MM}} = \text{ASPF}_{\text{BLENGTH}} - \text{ATPB}_{\text{ACTIVE}} \times B \quad (5)$$

where

- $\text{ASPF}_{\text{BLENGTH}}$ is the length of the allocation sizes or packet formats bitmap

$ATPB_{ACTIVE}$	is the total number of active access terminals (the number of ones in the access terminal presence bitmap)
B	is the number of bits per active access terminal in the allocation sizes or packet formats bitmap, where an indication of B is transmitted from the access network to the access terminal using the AllocationSizesOrPacketFormatsBitsPerActiveAT field of the GAM.

If R_{MM} is greater than 0, the last R_{MM} values of the allocation sizes or packet formats bitmap shall serve as a special resource availability bitmap, where each bit in the special resource availability bitmap corresponds to a resource. The last bit in the special resource availability bitmap corresponds to the last resource, the second to the last bit in the special resource availability bitmap corresponds to the second to the last resource, and so on. The access network shall set a bit in the special resource availability bitmap to '1' to indicate that the corresponding resource is currently being used and shall set a bit in the special resource availability bitmap to '0' to indicate that the corresponding resource is not currently being used.

8.6.5.5.1.6.4.6 Group resource allocation bitmap transmission

The GRA bitmap may be transmitted using the F-DCH or the F-SCCH. The access network may use the F-SCCH for transmitting the GRA bitmap if the size of the GRA bitmap, not including CRC bits, is less than or equal to 32 bits. The access network shall transmit an indication of whether the F-SCCH or F-DCH is used for transmitting the GRA bitmap using the BitmapChannelization field of the GAM. If the GRA frame is comprised of two PHY frames, the access network shall use the F-DCH for transmitting the GRA bitmap. The access network shall transmit a list of channel identifiers, which will be used to transmit the GRA bitmap to access terminals that require the GRA bitmap to determine a resource allocation using the BitmapChannelID field of the GAM. If the GRA frame occupies two PHY frames, the access network shall also transmit a list of PHY frames for each channel identifier, which will be used to transmit the GRA bitmap to access terminals that require the GRA bitmap to determine a resource allocation using the BitmapFrames field of the GAM. The FTC MAC shall instruct the Physical Layer that channelization (F-DCH or F SCCH), channel identifiers, and frames are to be used for GRA bitmap transmission.

8.6.5.5.1.6.4.7 Group resource allocation bitmap packet size

The packet size of the GRA bitmap, not including CRC bits, is defined as shown in Equation (6).

$$GRA_{SIZE} = \left\lceil \frac{RAB_{LENGTH} + RRF_{LENGTH} + ATPB_{LENGTH} + ASPFB_{LENGTH} + MMF_{LENGTH}}{16} \right\rceil \times 16 \quad (6)$$

where

RAB_{LENGTH}	is the length of the resource availability bitmap in units of bits
RRF_{LENGTH}	is the length of the reserved resources field in units of bits
$ATPB_{LENGTH}$	is the length of the access terminal presence bitmap in units of bits
$ASPFB_{LENGTH}$	is the length of the allocation sizes or packet formats bitmap in units of bits
MMF_{LENGTH}	is the length of the mini MAC field in units of bits

The length of the mini MAC field, in units of bits, is defined as

$$MMF_{LENGTH} = 1 + MMI_{LENGTH} + PF_{LENGTH}$$

where

MMI_{LENGTH}	is the length of the mini MAC index field in units of bits
PF_{LENGTH}	is the length of the packet format field in units of bits

Both MMI_{LENGTH} and PF_{LENGTH} have a maximum value of 4 bits. Each of these lengths is transmitted from the access network to the access terminal using the GAM. The quantities RAB_{LENGTH} , RRF_{LENGTH} , $ATPB_{LENGTH}$, $ASFPB_{LENGTH}$, and MMF_{LENGTH} shall be set to 0 if they are not defined in the Group Assignment Message.

If the GRA bitmap is transmitted on the F-DCH, the FTC MAC instructs the Physical Layer which packet size is to be used for GRA bitmap transmission.

8.6.5.5.1.1.6.4.8 Group resource allocation bitmap modulation

If the GRA bitmap is transmitted using the F-DCH, the access network shall transmit the GRA bitmap using the modulation indicated by the BitmapModulation field of the GAM. If the GRA bitmap is transmitted using the F-SCCH, the access network shall use QPSK modulation. The FTC MAC shall instruct the Physical Layer which modulation is to be used for GRA bitmap transmission.

8.6.5.5.1.1.6.5 Group resource allocation scrambling

If the GRA bitmap is transmitted using the F-DCH, the FTC MAC shall instruct the Physical Layer to use the GroupIndex for scrambling the GRA bitmap.

8.6.5.5.1.1.6.6 Position assignment

The access network may assign an access terminal a position within the group of access terminals. A position assignment is transmitted from the access network to the access terminal using the PositionIndex field of the GAM. The N^{th} position is associated with the N^{th} position in the concatenation of the resource availability bitmap and the access terminal presence bitmap, with the zeroth position corresponding to the left-most position. The access network may assign different positions to an access terminal in different group interlace offsets. If an access terminal is assigned a position in a particular group interlace offset that corresponds to one of the positions in the resource availability bitmap, the access network shall use the corresponding resource for transmissions to the access terminal during the group interlace offset.

Access terminals using residual resource assignment may be assigned to a group in order to monitor one or more GRA bitmaps, but not receive a position assignment. The position assignment for an access terminal may be omitted by setting NumConcurrentAssignments to '0' in the GAM.

8.6.5.5.1.1.6.7 Group interlace offset

The access network shall assign each access terminal for which a position is assigned to an group interlace offset. If the access network does not transmit a corresponding F-SPCH, the access network shall begin transmitting a packet (or a series of back-to-back packets) to the access terminal in its assigned group interlace offset. If the access network transmits a F-SPCH, with an 'ON' state, corresponding to the base node with the lowest index in the earliest frame within access terminal's allocation, the access network can begin transmitting a packet (or a series of back-to-back packets) in any group interlace offset. The group interlace offset is defined according to Equation (7).

$$GROUP\_INTERLACE\_OFFSET = \left\lfloor \frac{FRAME\_INDEX + SUPERFRAME\_INDEX \times N_{FLPHYFrames}}{N_{FRAMES\_PER\_GROUPINTERLACE}} \right\rfloor \bmod N_{GROUPINTERLACES} \quad (7)$$

where

FRAME_INDEX	is the index of the FL PHY frame within the superframe
SUPERFRAME_INDEX	is the index of the current superframe
$N_{FLPHYFrames}$	is the number of FL PHY frames that make up a superframe

$N_{\text{FRAMES\_PER\_GROUPINTERLACE}}$ is the number of FL PHY frames between successive occurrences of the same group interlace
 $N_{\text{GROUPINTERLACES}}$ is the total number of group interlaces

8.6.5.5.1.1.6.8 Concurrent assignments

The access network may assign an access terminal to more than one position within the group of access terminals, thereby allowing the access network to simultaneously transmit more than one packet to the access terminal. If the access network assigns two positions to an access terminal, the access network can configure the two positions in one of two ways. First, the access network can configure the position assignments to be completely independent of each other. Second, the access network can configure one position assignment to be HARQ extension of the other position assignment. In this case, the access network shall indicate the presence of HARQ transmissions 1 through *number_group_interlace_offsets* using the first position assignment and shall indicate the presence of HARQ transmissions *number_group_interlace_offsets*+1 to the maximum number of HARQ transmissions using the second position assignment. If more than one position index is assigned, the access network shall transmit an indication of the configuration using the ExtensionAssignment field of the GAM.

8.6.5.5.1.1.6.9 Mini MAC index assignment

The access network may assign an access terminal a mini MAC index. The mini MAC index is transmitted from the access network to the access terminal using the MiniMACIndex field of the GAM.

8.6.5.5.1.1.6.10 Persistent configurations

The access network can use one of the following four resource allocation schemes within the context of group resource allocation:

- First, the position assignment corresponding to the first transmission group interlace offset shall correspond to one of the bits in the resource availability bitmap; i.e., the resource is persistently allocated and the access terminal does not need to decode the group resource allocation bitmap for the first transmission group interlace offset. The position assignment(s) corresponding to other group interlace offsets shall correspond to one of the bits in the access terminal presence bitmap; i.e., the resources are not persistently allocated and the access terminal shall decode the group resource allocation bitmap to determine the resource allocated on those interlace offsets.
- Second, the position assignment corresponding to each of the group interlace offsets shall correspond to one of the bits in the resource availability bitmap; i.e., the resources are persistently allocated and the access terminal does not need to decode the group resource allocation bitmap for any group interlace offset.
- Third, the position assignment corresponding to the first transmission group interlace offset shall correspond to one of the bits in the access terminal presence bitmap. The resource allocated shall persist for subsequent HARQ transmission opportunities of the same packet. The access terminal does not need to decode the group resource allocation bitmap for group interlace offsets that do not correspond to the first transmission group interlace offset.
- Fourth, the position assignment corresponding to each of the group interlace offsets shall correspond to one of the bits in the access terminal presence bitmap.

8.6.5.5.1.1.6.11 Group switching

Upon receipt of a GAM with the AssignmentType field set to '01', the access terminal shall attempt to decode the GRA bitmap of the group specified by the AssignedGroupIndex field of the GAM in the next occurrence of the group interlace for which the AssignedGroupIndex is valid.

8.6.5.5.1.1.6.12 Group properties change

If the access terminal receives a GAM with the AssignmentType set to '11', and the access terminal is assigned to the group corresponding to the GroupIndex field, the access terminal shall determine its new position assignment as follows:

- If the original position assignment corresponds to one of the bits in the resource availability bitmap, the new position assignment is equal to the original position assignment.
- If the original position assignment corresponds to one of the bits in the access terminal presence bitmap, the access terminal shall determine its new position assignment according to Equation (8).

$$\text{Position}_{\text{NEW}} = \text{Position}_{\text{ORIG}} + (\text{RAB}_{\text{LENGTH,NEW}} - \text{RAB}_{\text{LENGTH,ORIG}}) \quad (8)$$

where

- Position_{ORIG} is the access terminal's position prior to receiving the group properties change
- RAB_{LENGTH,NEW} is the length of the resource availability bitmap in the GAM containing the group properties change
- RAB_{LENGTH,ORIG} is the length of the resource availability bitmap prior to receiving the GAM containing the group properties change

8.6.5.5.1.1.6.13 F-DCH transmission of GRA-based packets

8.6.5.5.1.1.6.13.1 Resource availability bitmap-based allocation

If the access network sets a bit in the resource availability bitmap to '1', the FTC MAC shall instruct the Physical Layer to transmit the packet, using the F-DCH, for the access terminal(s) assigned to the position corresponding to this bit using the resource corresponding to this bit.

8.6.5.5.1.1.6.13.2 Access terminal presence bitmap-based allocations

If the access network sets a bit in the access terminal presence bitmap to '1', the FTC MAC shall instruct the Physical Layer to transmit the packet(s), using the F-DCH, for the access terminal(s) assigned to the position corresponding to this bit using the N^{th} to $(N + NAR - 1)^{\text{th}}$ available resources within the set of available resources, where NAR is the number of allocated resources, and N is defined according to $N = RR + NRSB$, where RR is the number of resources that are indicated as being used by the reserved resources field (if present), and $NRSB$ is the number of resources that are allocated to access terminals with smaller positions that are within the access terminal presence bitmap. If an allocation sizes bitmap is not transmitted as part of the GRA bitmap, NAR is defined as one. If an allocation sizes bitmap is transmitted as part of the GRA bitmap, NAR is defined as the decimal equivalent of one more than the binary value indicated in the access terminal's portion of the allocation sizes bitmap.

The set of available resources (AR) is defined according to $AR = (R - RABR) - RAOG$, where R is the list of group resources, $RABR$ is the list of those resources that are indicated as being used in the resource availability bitmap, and $RAOG$ is the list of resources, within the group resources, that have been allocated to other groups.

8.6.5.5.1.1.6.13.3 Residual resource allocation

8.6.5.5.1.1.6.13.3.1 Mini MAC field-based allocation

If the mini MAC field is used, the FTC MAC shall instruct the Physical Layer to transmit the packet, using the F-DCH, for the access terminal corresponding to the mini MAC index field using the K^{th} to $(T - 1)^{\text{th}}$ available resources, subject to a maximum number of resources, where T is the total number of available

resources within the group resources, and K is defined according $K = RR + NR$, where RR is the number of resources that are indicated as being used by the reserved resources field (if present), and NR is the number of resources that are allocated to access terminals with positions that are within the access terminal presence bitmap.

The set of available resources (AR) is defined according to $AR = (R - RABR) - RAOG$, where R is the list of resources, $RABR$ is the list of those resources that are indicated as being used in the resource availability bitmap, and $RAOG$ is the list of resources that have been allocated to other groups.

If $T - K$ is greater than the value indicated in the MiniMACMaxResources field of the GAM, the FTC MAC shall instruct the Physical Layer to use the K^{th} to $K + \text{MiniMACMaxResources} - 1$ available resources for transmitting the packet to the access terminal indicated in the mini MAC field.

If the packet format field is included as part of the mini MAC field, the FTC MAC shall instruct the Physical Layer to use the packet format corresponding to the value indicated in the packet format field for transmitting the packet. If the packet format field is not included as part of the mini MAC field, the FTC MAC shall instruct the Physical Layer to use the packet format corresponding to the MiniMACDataPacketFormat field of the GAM for transmitting the packet.

8.6.5.5.1.1.6.13.3.2 F-SCCH-based allocation

The access network may assign an access terminal a channel node in a residual manner using the RRA-TD-FLAB, RRA-SCW-FLAB, or RRA-MCW-FLAB1 block types of the F-SCCH. For residual resource assignments, the FTC MAC shall instruct the physical layer to transmit the packet, using the F-DCH, using a set of residual resource assignment base nodes (RRA_{BN}) defined by Equation (9).

$$RRA_{BN} = (\text{CHANID}_{BN} - \text{GRA}_{BN}) \quad (9)$$

where

CHANID_{BN} is the list of base-nodes indicated in the F-SCCH

GRA_{BN} is the list of base-nodes that are indicated as being used by group resource allocation bitmaps known to the access terminal

The list of base nodes comprising GRA_{BN} contains those base-nodes that are indicated as being used by the resource availability bitmap, the reserved resources field, the access terminal presence bitmap, the allocation sizes bitmap, and any base-nodes used for transmitting the GRA bitmap itself. The available base-nodes may change for each HARQ transmission opportunity.

8.6.5.5.1.1.6.13.4 F-DCH scrambling

If the SharedPosition field of GAM is set to '1', the FTC MAC shall instruct the Physical Layer to use the GroupIndex for F-DCH scrambling. Otherwise, the FTC MAC shall instruct the Physical Layer to use the MACID of the targeted access terminal for scrambling.

8.6.5.5.1.1.6.14 Determining an allocation

8.6.5.5.1.1.6.14.1 Resource availability bitmap-based allocations

If an access terminal is assigned a position that corresponds to one of the positions within the resource availability bitmap, and the corresponding bit is set to '1', the access terminal shall determine its allocation as the resource that corresponds to this bit.

8.6.5.5.1.1.6.14.2 Access terminal presence bitmap-based allocations

If an access terminal is assigned a position that corresponds to one of the positions within the access terminal presence bitmap and the corresponding bit is set to '1', the access terminal shall determine its allocation as the N^{th} to $(N + NAR - 1)^{\text{th}}$ available resource within the set of available resources, where NAR is the number of allocated resources, and N is defined according to $N = RR + NRSB$, where RR is the number of resources that are indicated as being used by the reserved resources field (if present), and $NRSB$ is the number of resources that are allocated to access terminals with smaller positions that are within the access terminal presence bitmap. If an allocation sizes bitmap is not transmitted as part of the GRA bitmap, NAR is defined to be one. If an allocation sizes bitmap is transmitted as part of the GRA bitmap, NAR is defined as the decimal equivalent of one more than the binary value indicated in the access terminal's portion of the allocation sizes bitmap.

The set of available resources (AR) is defined according to $AR = (R - RABR) - RAOG$, where R is the list of group resources, $RABR$ is the list of those resources that are indicated as being used in the resource availability bitmap, and $RAOG$ is the list of resources, within the group resources, that have been allocated to other groups. The access terminal shall decode the GRA bitmap of other groups to determine $RAOG$ if the NumOtherGroupAllocationsRequiredToRead field of the GAM is greater than 0.

8.6.5.5.1.1.6.14.3 Residual resource allocation

8.6.5.5.1.1.6.14.3.1 Mini MAC field-based residual resource allocation

If the mini MAC field is used, the access terminal shall first determine if its MMI_{LENGTH} least significant bits of its assigned mini MAC index matches the MMI_{LENGTH} bits of mini MAC index field received as part of the GRA bitmap, where MMI_{LENGTH} is the length of the mini MAC index field. If a match is determined, the access terminal shall determine its allocation as the K^{th} to $T - 1^{\text{th}}$ available resources, subject to a maximum number of resources, where T is the total number of available resources within the group resources, and K is defined according to $K = RR + NR$, where RR is the number of resources that are indicated as being used by the reserved resources field (if present), and NR is the number of resources that are allocated to access terminals with positions that are within the access terminal presence bitmap.

The set of available resources (AR) is defined according to $AR = (R - RABR) - RAOG$, where R is the list of group resources, $RABR$ is the list of those resources that are indicated as being used in the resource availability bitmap, and $RAOG$ is the list of resources that have been allocated to other groups.

If $T - K$ is greater than the value indicated in the MiniMACMaxResources field of the GAM, the access terminal shall determine its allocation as the K^{th} to $K + \text{MiniMACMaxResources} - 1$ available resources.

If the packet format field is included as part of the mini MAC field, the access terminal shall process the received packet using the packet format corresponding to the value indicated in the packet format field. If the packet format field is not included as part of the mini MAC field, the access terminal shall process the received packet using the packet format corresponding to the MiniMACDataPacketFormat field of the GAM.

8.6.5.5.1.1.6.14.3.2 F-SCCH-based residual resource allocation

If the access terminal receives an F-SCCH containing a RRA-TD-FLAB, RRA-SCW-FLAB, or RRA-MCW-FLAB1 block type, the access terminal shall determine its allocation as the set of residual resource assignment base nodes (RRA_{BN}) defined by Equation (10).

$$RRA_{\text{BN}} = (\text{CHANID}_{\text{BN}} - \text{GRA}_{\text{BN}}) \quad (10)$$

where

- CHANID_{BN} is the list of base-nodes indicated in the RRA-TD-FLAB, RRA-SCW-FLAB, or RRA-MCW-FLAB1
- GRA_{BN} is the list of base-nodes that are indicated as being used by group resource allocation bitmaps known to the access terminal

The list of base nodes comprising GRA_{BN} contains those base nodes that are indicated as being used by the resource availability bitmap, the reserved resources field, the access terminal presence bitmap, the allocation sizes bitmap, and any base nodes used for transmitting the GRA bitmap. The available base nodes may change for each HARQ transmission opportunity.

8.6.5.5.1.1.7 Access terminal assignment management during handoff

If an *FLSSChanged* indication from the RCC MAC protocol is received, the access terminal shall clear all FL-ATAs and FL-NP-ATAs associated with the old FLSS.

If the access terminal receives an FLAB/NP-FLAB, that has the supplemental field set to '0', from the DFLSS, while the DFLSS is different from the FLSS, the access terminal shall issue a *ReverseControlChannelMAC.ChangeFLSS* command to change from the FLSS to the DFLSS. If the access terminal then receives an *FLSSChanged* indication from the RCC MAC Protocol, the access terminal shall process the *FLSSChanged* indication as specified above, and in addition shall update the appropriate FL-ATA/FL-NP-ATA according to the new FLAB/NP-FLAB.

The access terminal shall ignore all FLABs or NP-FLABs that come from sectors other than its current FLSS or its DFLSS.

8.6.5.5.1.2 Access terminal assignment processing

Access terminal assignment processing for persistent, non-persistent, and residual, and group resource assignments are described in 8.6.5.5.1.2.1 through 8.6.5.5.1.2.4. If a PDCAB is received from the FLCS MAC Protocol, and if both PDCABSupported and PDCABResourceSharingEnabled are set to '1', then this Protocol shall inform the Physical Layer of the set of F-SCCH blocks that are unused. The Physical Layer shall then demodulate the modulation symbols corresponding to these blocks if they are contained within an ATA. If a PDCAB block is not received, or if either PDCABSupported or PDCABResourceSharingEnabled is set to '0', all F-SCCH blocks are assumed to be occupied and the corresponding hop-ports are removed from the ATA.

8.6.5.5.1.2.1 Access terminal assignment processing for persistent assignments

The access terminal shall attempt to decode a MAC packet on each non-empty FL-ATA (contains one or more hop-ports).

If the packet format for the FL-ATA matches one of the configuration attributes FLMultiDecodeAssignedPFC, and if the size of the FL-ATA in base nodes is equal to the configuration attribute FLMultiDecodeAssignmentSizeK, then the access terminal shall try to decode multiple packet formats, as specified by configuration attributes FLMultiDecodeCompatiblePFCj where j goes from 0 to FLMultiDecodeNumCompatiblePFC – 1.

When an access terminal's FL-ATA changes (and is non-empty), the access terminal shall terminate any existing packet decoding and shall restart packet decoding attempts on the new ATA, unless one of the following conditions is satisfied:

- The FL-ATA change is due to decremental change from an overlapping non-persistent broadcast assignment.
- The FL-ATA change is due to a resource reassignment.
- The FL-ATA changes due to a HARQ extension.

The access terminal shall continue packet decoding as usual in the above cases.

If a Lower MAC packet on the FL-ATA is successfully decoded, as indicated by the PHY, the access terminal shall transmit a positive ACK via the RCC MAC to the access network. Refer to 8.1.5.1 for detailed acknowledgment timing. The payload of the packet is then passed up to the Packet Consolidation Protocol for further processing.

If the access terminal receives an F-SPCH from the FLCS MAC Protocol with a value of '1' for a FL-ATA, then the access terminal shall begin to decode assuming the frame in which this F-SPCH is sent is the start of the packet.

The access terminal shall expire an ATA if a packet on that FL-ATA fails to decode successfully for $N \times M$ consecutive HARQ attempts, and it has not received any F-SPCH indication with a value of '1' during that period, where $N = \text{NumFailPacketsforDrop}$, and M is the maximum number of HARQ transmissions (see 8.6.5.7). The Physical Layer Protocol determines whether or not a packet is successfully decoded.

If a Lower MAC packet on the FL-ATA is successfully decoded, or if the access terminal determines it has failed, and the number of packets that have failed is less than $\text{NumFailPacketsforDrop}$, then the following is true:

- If StartofPacketMode is set to '0', then the access terminal shall begin decoding the next packet on the next assigned PHY frame for the FL-ATA.
- If StartofPacketMode is set to '1', then the access terminal shall begin to decode the next packet on the next PHY frame that meets the criteria that it is $m \times \text{StartofPacketFrame}$ HARQ transmissions from the frame the assignment was sent, where m is any integer, unless it receives an F-SPCH message indicating that it should begin decoding sooner.

8.6.5.5.1.2.2 Access terminal assignment processing for non-persistent assignments

For each non-empty FL-NP-ATA_{UC} , the access terminal shall attempt to decode a Lower MAC packet. If a Lower MAC packet on the FL-NP-ATA_{UC} is successfully decoded, as indicated by the PHY, the access terminal shall reset the supervision timer and transmit a positive ACK via the RCC MAC to the access network. The access terminal may choose to expire its FL-NP-ATA_{UC} at this time if it is not performing ACK to NACK detection. Refer to 8.1.5.1 for detailed acknowledgment timing. The payload of the packet is then passed up to the Packet Consolidation Protocol for further processing.

For each non-empty FL-NP-ATA_{BC} , the access terminal shall attempt to decode a Lower MAC packet. If a Lower MAC packet on the FL-NP-ATA_{BC} is successfully decoded, as indicated by the Physical Layer, the access terminal shall expire the FL-NP-ATA_{BC} . The access terminal shall not transmit any acknowledgements on the R-ACKCH for packets received on an FL-NP-ATA_{BC} .

If not already expired, the access terminal shall expire an NP-ATA after the final HARQ retransmission.

8.6.5.5.1.2.3 Access terminal assignment processing for residual resource assignments

For each non-empty FL-R-ATA, the access terminal shall attempt to decode a Lower MAC packet. If a Lower MAC packet on the FL-R-ATA is successfully decoded, as indicated by the PHY, the access terminal shall reset the supervision timer and transmit a positive ACK via the RCC MAC to the access network. The access terminal may choose to expire its FL-R-ATA at this time if it is not performing ACK to NACK

detection. Refer to 8.1.5.1 for detailed acknowledgment timing. The payload of the packet is then passed up to the Packet Consolidation Protocol for further processing.

If not already expired, the access terminal shall expire an FL-R-ATA after the final HARQ retransmission.

8.6.5.5.1.2.4 Access terminal assignment processing for group resource assignments

For each non-empty FL-GR-ATA, the access terminal shall attempt to decode a Lower MAC packet. For multicast FL-GR-ATAs, the access terminal should attempt to decode these with a GroupIndex rather than a MACID. If a Lower MAC packet on the FL-GR-ATA is successfully decoded, as indicated by the PHY, the access terminal shall reset the supervision timer and transmit a positive ACK via the RCC MAC to the access network. The access terminal may choose to expire its FL-GR-ATA at this time if it is not performing ACK to NACK detection and if none of the following two conditions apply:

- The BackToBack field of the GAM for the access terminal is set to '1'.
- The PersistentConfiguration field of the GAM for the access terminal is set to '01'.

Refer to 8.1.5.1 for detailed acknowledgment timing. The payload of the packet is then passed up to the Packet Consolidation Protocol for further processing.

If not already expired, the access terminal shall expire an FL-GR-ATA after the final HARQ transmission if none of the following two conditions apply:

- The BackToBack field of the GAM for the access terminal is set to '1' and a positive ACK has been transmitted by the access terminal corresponding to the transmitted packet on a FL-GR-ATA.
- The PersistentConfiguration field of the GAM for the access terminal is set to '01'.

8.6.5.5.1.3 Header processing

If a Lower MAC packet using MACID_{PAGE} is successfully decoded, as indicated by the PHY, the packet shall be processed according to the following rules:

- a) The access terminal shall parse the packet according to the header and trailer specified for transmissions using MACID_{PAGE}.
- b) The access terminal shall discard the consolidated packet.
- c) If the PagingID field in the Lower MAC layer header is equal to any entry or the least significant bits of any entry of the Route Control Protocol's PagingIDList, then this protocol shall return an *PageReceived* indication. If extra information was included in the page, the *PageReceived* indication shall be accompanied by the following information:
 - If a BaseAccessSequence was included in the page and EnhancedPageFeaturesSupport is set to 0x01, the indication shall be accompanied by a PageResponseAccessSequence = BaseAccessSequence + n – 1, where the matching PagingID was the nth listed PagingID in the header.
 - If a AccessPersistenceClass was included in the page and EnhancedPageFeaturesSupport is set to 0x01, the indication shall be accompanied by the included AccessPersistenceClass.

8.6.5.5.1.4 Supervision procedures

The access terminal shall return a *SupervisionFailed* indication if it sends NULL CQI to the FLSS for a continuous period of duration exceeding FTCSupervisionTimeout.

8.6.5.5.2 Access network requirements

The access network shall keep a variable NumLayers for each SCW/MCW FL-ATA/FL-NP_ATA/FL-R-ATA. On transmission of any type of SCW FLAB (including RRA-SCW-FLAB etc.) via the FLCS MAC protocol, NumLayers for that FL-ATA shall be set according to the FL Pilot/MIMO field of the SCW-FLAB. On transmission of any type of MCW-FLAB (including RRA-MCW-FLAB etc.) via the FLCS MAC Protocol, NumLayers shall be set to the number of non-zero packet formats given in the MCW-FLAB.

8.6.5.5.2.1 Access network assignment management

Upon receiving a *TunedAway* indication from the Air Link Management Protocol, the access network shall expire all the FL-ATAs, FL-NP-ATAs, and FL-R-ATAs for that access terminal.

If the FLNumSDMADimensions > 1, then the access network shall ensure that assignments to different access terminals that contain hop-ports that map to the same subcarriers have the same F-DPICH format, as described in 8.6.5.7.

8.6.5.5.2.2 F-DCH transmissions associated with persistent assignments

The access network may formulate and transmit a Lower MAC packet on a particular set of PHY frames according to the FL-ATA and the PF selected for transmission. The Lower MAC shall pass to the Physical Layer Protocol the set of hop-ports specified by the FL-ATA on the appropriate PHY frames. The access network shall inform the access terminal of its assignment using signaling blocks as described in 8.5.5.4.1.1.1.

If the packet format sent for the FL-ATA matches one of the configuration attributes FLMultiDecodeAssignedPFK, and if the size of the FL-ATA in base nodes is equal to configuration attribute FLMultiDecodeAssignmentSizeK, then the access network may transmit one of multiple packet formats, as specified by configuration attributes FLMultiDecodeCompatiblePFKj where j goes from 0 to FLMultiDecodeNumCompatiblePFK – 1.

If TuneAwayStatus is equal to '1', then the access network shall not send any Lower MAC packets on any ATAs to this access terminal.

If a positive ACK is received that corresponds to the transmitted packet, transmission of the Lower MAC packet shall terminate. (Note that for multi-codeword assignments assigned by MCW-FLABs, the ACKs for all the layers need to be received before the transmission of the Lower MAC packet terminates.)

If StartofPacketMode is equal to '0', then the following are true:

- Once an ACK is received, the access network may immediately send the next packet, if it is available, or it may wait any integer multiple of the maximum number of HARQ transmissions before sending the next packet, as long as the FL-ATA does not expire due to the rules discussed in this subclause.
- If no ACK is received after the maximum number of HARQ transmission of a packet, the access network may send the next packet on the next HARQ transmission, or it may wait any integer multiple of the maximum number of HARQ transmissions before sending the next packet, as long as the FL-ATA does not expire due to the rules specified below.
- The access network may send a new packet any time if it sends a message through F-SPCH indicating that it is doing so.

If StartofPacketMode is equal to '1', then the following are true:

- The access network may send a packet on any multiple of StartofPacketFrame HARQ transmissions from the first HARQ transmission of the assignment. (The frame in which the assignment was sent.)

- The access network may send a new packet any time if it sends a message through F-SPCH indicating that it is doing so.

If the access network transmits a packet for the maximum number of HARQ transmissions of the PF selected for this Lower MAC packet without receiving an ACK, and this is the N^{th} time that this has happened, where $N = \text{NumFailPacketsDrop}$, the access network shall expire the FL-ATA, and shall return a *ForwardTrafficPacketsMissed* indication along with parameters that uniquely identify the lost packet. The method of uniquely identifying the packet is out of the scope of this standard.

The access network should send no more than one Lower MAC packet per FL-ATA. The access network shall not send packets to the same access terminal over two different FL-ATAs if they overlap in time.

If no packet is available for transmission in a given PHY frame for a particular FL-ATA, the access network shall not send anything on the FL-ATA.

If *StartofPacketMode* is equal to '0', and if no packet is available for transmission for a particular FL-ATA for a period larger than or equal to the maximum number of HARQ transmissions, then the access network should count each group of maximum number of HARQ transmission opportunities as one “failure” when comparing the number of failures to *NumFailPacketsDrop* in order to determine if it should drop the FL-ATA.

If *StartofPacketMode* is equal to '1', and the access network is not currently transmitting a packet, and if no packet is available for transmission on a designated start of packet FL-PHY frame, and no F-SPCH message is sent in the next six HARQ retransmissions to indicate the start of a new packet, then the access network should consider this a “failure” when it compares the number of failures to *NumFailPacketsDrop* in order to determine if it should drop the FL-ATA.

8.6.5.5.2.3 F-DCH transmissions associated with non-persistent assignments and residual resource assignments

The access network may formulate and transmit a Lower MAC packet on a particular set of PHY frames to the NPATA/R-ATA and the PF selected for transmission. This protocol shall pass to the Physical Layer Protocol the set of hop-ports specified by the FL-NP-ATA/FL-R-ATA. The access network shall not start an F-DCH transmission containing a page in the first two frames of a superframe. (A page can be sent using an FTC MAC packet transmission using a reserved $\text{MACID}_{\text{PAGE}}$ wherein the Lower MAC header includes the *PagingID* field. See 8.6.6.2.)

The number of HARQ retransmissions over a FL-NP-ATA is less than or equal to the maximum number of HARQ transmission, unless the FL-NP-ATA has been extended for additional HARQ retransmissions via HARQ extension message/messages. The number of HARQ retransmissions over a FL-R-ATA is always less than or equal to the maximum number of HARQ transmission.

If *TuneAwayStatus* is equal to '1', then the access network shall not send any Lower MAC packets on any $\text{FL-NP-ATA}_{\text{UCS}}$ to this access terminal. Lower MAC packets being sent on $\text{FL-NP-ATA}_{\text{BCS}}$ may still be sent.

The access network should send no more than one Lower MAC packet over a $\text{FL-NP-ATA}_{\text{UC}}$. The access network shall not send packets to the same access terminal over two different $\text{FL-NP-ATA}_{\text{UCS}}$ if they overlap in time. The access network should send no more than one Lower MAC packet over a $\text{FL-NP-ATA}_{\text{BC}}$. The access network shall not send packets over two different $\text{FL-NP-ATA}_{\text{BCS}}$ if they overlap in time. The access network should not send Lower MAC packets to the same MACID over both an FL-ATA and a $\text{FL-NP-ATA}_{\text{UC}}$ if they overlap in time. The access network should send no more than one Lower MAC packet over a FL-R-ATA.

If a positive ACK is received that corresponds to the transmitted packet on a FL-NP-ATA_{UC}/FL-R-ATA, transmission of the Lower MAC packet shall terminate and the access network shall expire the FL-NP-ATA_{UC}. For multi-codeword assignments assigned by NP-MCW-FLABs/RRA-MCW-FLABs, the ACKs for all the layers need to be received before the transmission of the Lower MAC packet terminates. If no ACK is received for this packet after maximum number of HARQ transmissions, the access network shall return a *ForwardTrafficPacketsMissed* indication along with parameters that uniquely identify the lost packet. The method of uniquely identifying the packet is out of the scope of this standard.

FL-NP-ATAs/FL-R-ATAs shall be expired after the final HARQ transmission, if not already expired.

8.6.5.5.2.4 F-DCH transmissions associated with group resource assignments

The access network may formulate and transmit a Lower MAC packet on a particular set of PHY frames to the FL-GR-ATA and the PF selected for transmission. This protocol shall pass to the Physical Layer Protocol the set of hop-ports specified by the FL-GR-ATA.

The access network shall use a GroupIndex rather than a MACID to send packets on multicast FL-GR-ATAs. The GroupIndex is specified by the GAM.

If a positive ACK is received that corresponds to the transmitted packet on a FL-GR-ATA, transmission of the Lower MAC packet shall terminate, and the access network shall expire the FL-GR-ATA if none of the following two conditions apply:

- The BackToBack field of the GAM for the corresponding access terminal is set to '1'.
- The PersistentConfiguration field of the GAM for the corresponding access terminal is set to '01'.

If no ACK is received for this packet after maximum number of HARQ transmissions, the access network shall return a *ForwardTrafficPacketsMissed* indication along with parameters that uniquely identify the lost packet. The method of uniquely identifying the packet is out of the scope of this standard.

If not already expired, the FL-GR-ATAs shall be expired after the final HARQ transmission if none of the following two conditions apply:

- The BackToBack field of the GAM for the corresponding access terminal is set to '1' and a positive ACK is received that corresponds to the transmitted packet on a FL-GR-ATA.
- The PersistentConfiguration field of the GAM for the corresponding access terminal is set to '01'.

Note that multicast FL-GR-ATAs are not acknowledged and expire after the maximum number of HARQ transmissions.

8.6.5.6 Channel trees

A channel tree is used to specify sets of hop-ports that are associated with each node identification number (NodeID). A set of hop-ports is said to be “mapped to a node” and a node “maps” a set of hop-ports. The following terms are used in association with channel trees in this standard:

hop-port The fundamental unit of resource assignment. Each hop-port maps to one unique subcarrier. Hop-ports from different subtrees may map to the same subcarrier. Also, multiple hop-ports from the same subtree may map to the same subcarrier when ResourceChannelMuxMode is equal to 1. The mapping of hop-ports to subcarriers varies with time according to hopping rules specified in the Physical Layer specification.

node A node corresponds to a single NodeID.

children, descendants	Nodes that map a subset of the hop-ports mapped by a node.
parents, ancestors	Nodes that map a superset of the hop-ports mapped by a node.
base-nodes	Nodes with no children. Base-nodes are assigned specific Physical Layer resources—in this case, hop-ports.
MinHopPortsPerNode	The minimum number of hop-ports mapped to a node. MinHopPortsPerNode equals 16 in this standard.

The indices of hop-ports follows a convention used throughout the Lower MAC layer. Namely, hop-port indices are allocated sequentially starting from index 0. The base nodes are ordered in a left-to-right ordering as drawn, and hop-port indices are allocated sequentially starting from index 0 to the base nodes from left to right.

Because nodes define orthogonal assignments, the use of a node in the tree can restrict use of other nodes. Thus, if a node is in use, then all descendants and ancestors of the node are unavailable for use and are called “restricted” nodes.

The channel tree shall be described as a function of the variable N_{FFT} , which shall be given by the TotalNumSubcarriers field of the OMP.

The set of hop-ports specified by a NodeID shall be the union of usable hop-ports mapped by all base-nodes that are descendants of the node specified by NodeID. “Usable hop-ports” are defined by the Physical Layer Protocol.

If DuplexingMode = TDD63, the hop-ports corresponding to pairs of FL PHY frames indexed $2k$ and $2k + 1$ (the first FL PHY frame of each pair has an even FL PHY frame index) are mapped to only one channel tree. This is done by mapping two successive base nodes that have a common parent to the same set of hop-ports of the two different FL PHY frames, such that the base node with the lower index maps to the FL PHY frame with the lower index. In order to have a uniform treatment of all the cases, the variable FLFrameMergeFactor is defined such that $FLFrameMergeFactor = 2$ if DuplexingMode = TDD63 and $FLFrameMergeFactor = 1$ if DuplexingMode $\neq$ TDD63.

In the case of SDMA transmission, multiple modulation symbols are multiplexed over the same physical subcarriers at different spatial dimensions. For each spatial dimension, an SDMA subtree is defined to specify all the associated hop-ports. The SDMA order for forward link transmissions, Q_{SDMA} , is equal to FLNumSDMADimensions, which is public data of the OMP.

The Overhead Messages Protocol defines the forward link ResourceChannelMuxMode and the partition of resources between DRCH and BRCH. Note that the channel tree topology is fixed regardless of the ResourceChannelMuxMode, where a total of $FLFrameMergeFactor \times N_{FFT} / \text{MinHopPortsPerNode}$ base nodes are defined for each SDMA dimension.

The mapping of nodes in the channel tree to hop-ports, however, depends on the ResourceChannelMuxMode. The number of hop-ports is equal to $Q_{SDMA} N_{HOP-PORTS-PER-SUBTREE}$, where $N_{HOP-PORTS-PER-SUBTREE} = FLFrameMergeFactor \times (N_{DRCH} + N_{FFT})$ in ResourceChannelMuxMode 1 and equal to N_{FFT} in ResourceChannelMuxMode 2. Here N_{DRCH} is equal to $N_{SUBZONE, MAX} \times N_{DRCH-SUBZONES}$, where $N_{SUBZONE, MAX}$ is equal to FLSubzoneSize and $N_{DRCH-SUBZONES}$ is equal to NumDRCHSubzones. FLSubzoneSize and NumDRCHSubzones are parameters of the OMP.

The set of hop-ports shall be divided into Q_{SDMA} “SDMA subtrees,” indexed 0 through $Q_{SDMA} - 1$. Each SDMA subtree shall have the same number of hop-ports, and the SDMA subtree with index q shall contain hop-ports indexed $q \times N_{HOP-PORTS-PER-SUBTREE}$ through $(q + 1) \times N_{HOP-PORTS-PER-SUBTREE} - 1$.

SDMA subtree with index 0 is illustrated in Figure 58 for $N_{FFT} = 512, 1024, \text{ and } 2048$ and $FLFrameMergeFactor = 1$. For $N_{FFT} = X$, all nodes above the dashed line marked with $N_{FFT} = X$ are included in the channel tree. One deassignment NodeID, $NodeID_{DEASSIGN}$, is set to $N_{FFT}/(\text{MinHopPortsPerNode}/2) - 1$. Other SDMA subtrees are identical as the SDMA subtree 0, but with unique NodeID to hop-ports mapping.

Note that the channel tree shown in Figure 58 ensures a single parent node for $FLSubzoneSize$ hop-ports.

The mapping of hop-ports to each base-node when $ResourceChannelMuxMode = 2$ is described as follows. Each base-node maps to $\text{MinHopPortsPerNode}$ hop-ports, the first $\text{MinHopPortsPerNode}$ hop-ports (indices 0 to $\text{MinHopPortsPerNode} - 1$) to the base-node with the lowest NodeID, the second $\text{MinHopPortsPerNode}$ hop-ports to the next base-node, etc., until all hop-ports are mapped for all SDMA subtrees. See Table 101 for the mapping for $N_{FFT} = 512$, $\text{MinHopPortsPerNode} = 16$, and $FLFrameMergeFactor = 1$, and $Q_{SDMA} = 4$ in $ResourceChannelMuxMode 2$. Also, see Table 103 and Figure 59 for the mapping $N_{FFT} = 512$, $\text{MinHopPortsPerNode} = 16$, $FLFrameMergeFactor = 2$, and $Q_{SDMA} = 1$.

In $ResourceChannelMuxMode 1$, the first $FLFrameMergeFactor \times N_{DRCH}$ DRCH hop-ports are mapped to the first $FLFrameMergeFactor \times N_{DRCH}/\text{MinHopPortsPerNode}$ base nodes in the SDMA subtree 0 and the next $FLFrameMergeFactor \times N_{FFT} - N_{DRCH}$ hop-ports are mapped to the next $FLFrameMergeFactor \times (N_{FFT} - N_{DRCH})/\text{MinHopPortsPerNode}$ base nodes in SDMA subtree 0. Define $B = FLFrameMergeFactor \times N_{FFT}/\text{MinHopPortsPerNode}$ and $D = FLFrameMergeFactor \times N_{DRCH}/\text{MinHopPortsPerNode}$. For each hop-port index j , compute $BRCHIndexForHopport(j)$ as $\lfloor j/\text{MinHopPortsPerNode} \rfloor - D$. The $FLFrameMergeFactor \times N_{FFT}$ hop-ports are mapped to $B-D$ channel-tree nodes by not assigning channel-tree base nodes to equally spaced instances of $BRCHIndexForHopport(j)$, namely the ones that satisfy $BRCHIndexForHopport(j) = \lceil l \times B/D \rceil$ for some value of l in the range $0 \leq l < D - 1$. The hop-ports corresponding to these instances are merged into the next higher channel tree node.

To be precise, the channel tree node (note that this node is always a base-node) corresponding to hop-port index j shall be given by

$$\left\lfloor \left\lfloor \frac{j - FLFrameMergeFactor \times N_{DRCH}}{\text{MinHopPortsPerNode}} \right\rfloor \frac{FLFrameMergeFactor \times N_{FFT} - N_{DRCH}}{FLFrameMergeFactor \times N_{FFT}} \right\rfloor + \frac{FLFrameMergeFactor \times N_{DRCH}}{\text{MinHopPortsPerNode}} + \frac{FLFrameMergeFactor \times N_{FFT}}{\text{MinHopPortsPerNode}} - 1$$

See Table 102 for the mapping for $N_{FFT} = 512$, $\text{MinHopPortsPerNode} = 16$, $FLSubzoneSize = 64$, $\text{NumDRCHSubzones} = 4$, and $Q_{SDMA} = 2$ in $ResourceChannelMUXMode 1$.

**Table 101—FL base node NodeID and subtree index to hop-port for
ResourceChannelMUXMode 2**(N_{FFT}=512, MinHopPortsPerNode=16, FLFrameMergeFactor = 1 and Q_{SDMA}=4)

(Base node NodeID, subtree Index)	Hop-ports mapped
(31, 0)	0–15
(32, 0)	16–31
—	—
(62, 0)	496–511
(31, 1)	512–527
(32, 1)	528–543
—	—
(62, 1)	1008–1023
—	—
(62, 3)	2032–2047

**Table 102—FL base node NodeID and subtree index to hop-port for
ResourceChannelMUXMode 1**(N_{FFT}=512, MinHopPortsPerNode=16, FLSubzoneSize = 64, NumDRCHSubzones = 4 and Q_{SDMA}=2)

(Base node NodeID, subtree Index)	Hop-ports mapped
(31, 0)	0–15
—	—
(46, 0)	240–255
(47, 0)	256–287
...	...
(62, 0)	736–767
(31, 1)	768–784
—	—
(62, 1)	1504–1535

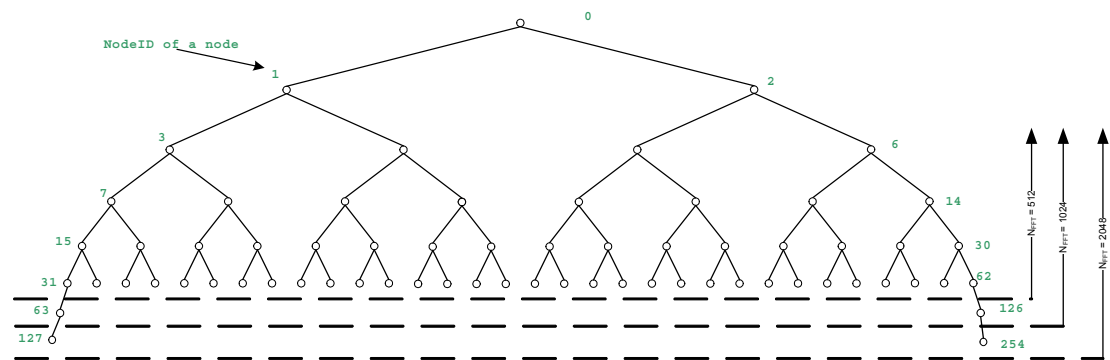


Figure 58—Forward Link SDMA subtree index 0

Table 103—FL base node NodeID and subtree index to hop-port for
ResourceChannelMUXMode=2
($N_{FFT}=512$, MinHopPortsPerNode=16, FLFrameMergeFactor = 2, and $Q_{SDMA}=1$)

(Base node NodeID, subtree Index)	(Hop-ports mapped, FL PHY frame index)
(63, 0)	(0–15, 2k)
(64, 0)	(0–15, 2k + 1)
(65, 0)	(16–32, 2k)
(66, 0)	(16–32, 2k + 1)
—	—
(93, 0)	(240–255, 2k)
(94, 0)	(240–255, 2k + 1)
(95, 0)	(256–267, 2k)
(96, 0)	(256–267, 2k + 1)
—	—
(126, 0)	(496–511, 2k + 1)

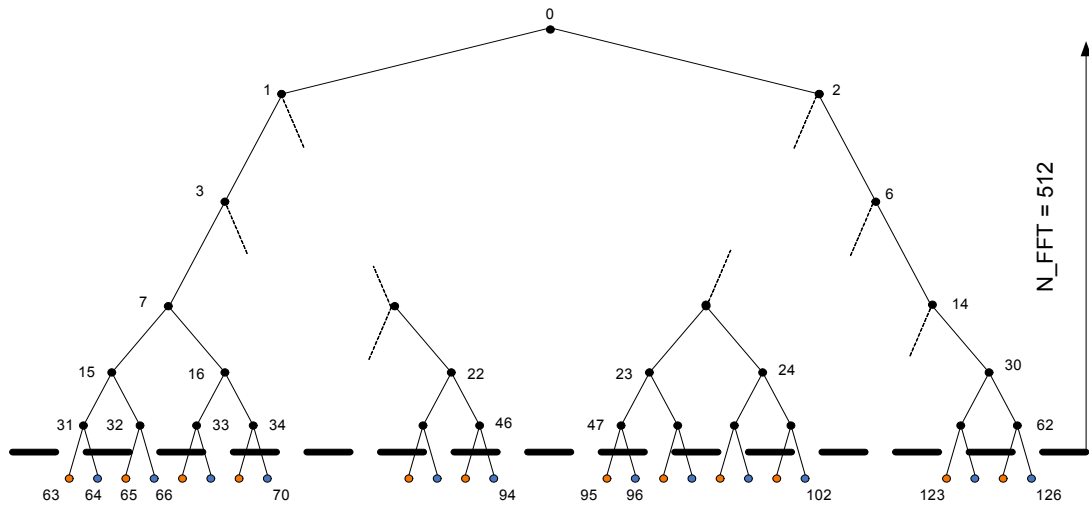


Figure 59—FL SISO Channel Tree for DuplexingMode=TDD63 ResourceChannelMUXMode=2
($N_{FFT}=512$, MinHopPortsPerNode=16, FLFrameMergeFactor = 2)

8.6.5.7 Packet formats

A packet format (PF) specifies the spectral efficiency, maximum number of transmissions, and the modulation format to be used for each transmission of a data packet. Each packet format is indexed by a packet format index. The modulation format is specified by the number of bits in each modulation symbol, which is denoted by modulation order. Modulation orders of 2, 3, 4, and 6 correspond to QPSK, 8PSK, 16-QAM, and 64-QAM modulations, respectively. These modulations are described in detail in the Physical Layer specification. The size of the Lower MAC packet that is provided to the Physical Layer is a function of the packet format as well as the set of hop-ports that are assigned to the data packet (to be transmitted on the F-DCH Physical Layer channel). The packet size is byte aligned. Also, a CRC of length $N_{CRC,Data}$ is added by the Physical Layer for F-DCH packets, where $N_{CRC,Data}$ is a constant of the Physical Layer Protocol. Hop-ports that are assigned to the FLCS shall be removed from the set of usable hop-ports for packet size computation. The FLCS hop-ports are defined by the Physical Layer Protocol.

8.6.5.7.1 SISO mode packet formats

The packet format consists of four bits. These four bits index the spectral efficiency and the modulation format to be used for each HARQ transmission of a data packet. The indexing is described in Table 104. Note that the default maximum number of transmission is six; however, a persistent assignment could be extended to more than six transmissions through an FLAB.

The packet size for each assignment is computed based on the spectral efficiency listed in Table 104 and the assignment size. Let s denote the spectral efficiency of the packet format at the first HARQ transmission (specified in the packet format table); let H denote the number of usable hop-ports in the assignment at the first HARQ transmission; let N denote the number of information bits in the packet excluding CRC. The formulae below refer to all the cases except the one when DuplexingMode = TDD63. For all assignments except for the BRCH assignments in ResourceChannelMUXMode 1, and if DuplexingMode $\neq$ TDD63, the packet size is given by

$$N = f_0(s, H), \text{ where } f_0(s, H) = 8 \lfloor sH \rfloor - N_{CRC,Data}$$

for regular assignments, and

$$N = f_1(s, H), \text{ where } f_1(s, H) = 8 \lfloor 3sH \rfloor - N_{\text{CRC,Data}}$$

for extended frame assignments, respectively.

For all BRCH assignments in ResourceChannelMuxMode 1, and if DuplexingMode $\neq$ TDD63, the packet size is given by

$$N = f_2(s, H), \text{ where } f_2(s, H) = 8 \left\lfloor \frac{sH(N_{\text{FFT}} - N_{\text{DRCH}})}{N_{\text{FFT}}} \right\rfloor - N_{\text{CRC,Data}}$$

for regular assignments, and

$$N = f_3(s, H), \text{ where } f_3(s, H) = 8 \left\lfloor \frac{3sH(N_{\text{FFT}} - N_{\text{DRCH}})}{N_{\text{FFT}}} \right\rfloor - N_{\text{CRC,Data}}$$

for extended frame assignments, respectively.

If DuplexingMode = TDD63, then let s denote the spectral efficiency of the packet format at the first HARQ transmission (specified in the packet format table); let H_0 denote the number of usable hop-ports in the assignment at the first HARQ transmission on the even-indexed FL PHY frame and H_1 denote the number of usable hop-ports in the assignment at the first HARQ transmission on the odd-indexed FL PHY frame.

Compute the numbers N_0 , N_1 , and N_2 using the appropriate function f_p where p can be 0, 1, 2, or 3 function of the assignment size: $N_0 = f_p(s, H_0)$, $N_1 = f_p(s, H_1)$, $N_2 = f_p(s, H_0 + H_1)$.

If $N_2 < \text{MaxPHYSubPacketSize}$, where MaxPHYSubPacketSize is a PHY protocol constant, then the packet size shall be N_2 and each transmission of the packet will be over two PHY frames. Otherwise, two F-DCH packets of sizes N_0 and N_1 shall be generated, which will be encoded separately and transmitted over even-indexed and odd-indexed PHY frames, respectively.

Table 104—Forward Link Packet Formats—SISO Mode

Packet Format Index	Spectral efficiency on 1 st transmission	Modulation order for each transmission					
		1	2	3	4	5	6+
0	0.33	2	2	2	2	2	2
1	0.67	2	2	2	2	2	2
2	0.94	2	2	2	2	2	2
3	1.5	4	3	3	3	3	3
4	2.0	4	3	3	3	3	3
5	2.5	6	4	4	3	3	3
6	3.0	6	4	4	4	4	4
7	3.5	6	4	4	4	4	4

Table 104—Forward Link Packet Formats—SISO Mode (continued)

Packet Format Index	Spectral efficiency on 1 st transmission	Modulation order for each transmission					
		1	2	3	4	5	6+
8	4.0	6	6	4	4	4	4
9	4.5	6	6	4	4	4	4
10	5.0	6	6	4	4	4	4
11	6.0	6	6	4	4	4	4
12	7.0	6	6	6	4	4	4
13	8.0	6	6	6	4	4	4
14	9.5	6	6	6	6	4	4
15	NULL						

PF index 15 is used to indicate NULL, or no PF, and is useful in MCW assignments that utilize fewer than the maximum number of layers for transmission (see 8.5.5.4.1.1.1).

8.6.5.7.1.1 Group resource allocation packet formats

For group resource allocation, the spectral efficiency (s) for packet formats 2–4 can be offset with a value s_{offset} using the SpectralEfficiencyOffset field of the group assignment message. A modified spectral efficiency (s_{mod}) is given by $s + s_{\text{offset}}$. For packet formats 2–4, s_{mod} replaces the value of s in the previous equations. Similarly, for packet formats 3–4, the modulation order can be adjusted to QPSK for all HARQ transmissions using the ModulationOffset field of the GAM. If the ModulationOffset field of the GAM is set to '0', the modulation orders from Table 104 are used, and if the ModulationOffset field of the GAM is set to '1', QPSK modulation is used.

8.6.5.7.2 MIMO multi-codeword mode packet formats

The packet format for each layer of the MIMO multi-codeword mode transmission consists of four bits. The indexing has the same format as SISO mode packet format as shown in Table 104. The packet size for each layer is computed in the same way as SISO mode packet size as explained in 8.6.5.7.1.

8.6.5.7.3 MIMO single-codeword mode packet formats

The packet format for the MIMO single-codeword mode transmission consists of four bits. The indexing has the same format as SISO mode packet format as shown in Table 104.

The packet size for each assignment is a computed based on the spectral efficiency listed in the Table 104, the rank of the transmission and the assignment size. Let s denote the spectral efficiency of the packet format at the first HARQ transmission; let r denote the rank; let H denote the number of usable hop-ports in the assignment at the first HARQ transmission; let N denote the number of information bits in the packet. For all assignments expect for the BRCH assignments in ResourceChannelMuxMode 1, and if DuplexingMode takes a different value than TDD63, the packet size is given by

$$N = f_0(s, H), \text{ where } f_0(s, H) = 8 \lfloor srH \rfloor - N_{\text{CRC,Data}}$$

for regular assignments, and

$$N = f_1(s, H), \text{ where } f_1(s, H) = 8 \lfloor 3srH \rfloor - N_{\text{CRC,Data}}$$

for extended frame assignments, respectively.

For all BRCH assignments in ResourceChannelMuxMode 1, and if DuplexingMode takes a different value than TDD63, the packet size is given by

$$N = f_2(s, H), \text{ where } f_2(s, H) = 8 \left\lfloor \frac{srH(N_{\text{FFT}} - N_{\text{DRCH}})}{N_{\text{FFT}}} \right\rfloor - N_{\text{CRC,Data}}$$

for regular assignments, and

$$N = f_3(s, H), \text{ where } f_3(s, H) = 8 \left\lfloor \frac{3srH(N_{\text{FFT}} - N_{\text{DRCH}})}{N_{\text{FFT}}} \right\rfloor - N_{\text{CRC,Data}}$$

for extended frame assignments, respectively.

If DuplexingMode = TDD63, then let s denote the spectral efficiency of the packet format at the first HARQ transmission (specified in the packet format table); let H_0 denote the number of usable hop-ports in the assignment at the first HARQ transmission on the even-indexed FL PHY frame and H_1 denote the number of usable hop-ports in the assignment at the first HARQ transmission on the odd-indexed FL PHY frame.

Compute the numbers N_0 , N_1 , and N_2 using the appropriate function f_p where p can be 0, 1, 2, or 3 function of the assignment size $N_0 = f_p(s, H_0)$, $N_1 = f_p(s, H_1)$, $N_2 = f_p(s, H_0 + H_1)$.

If $N_2 < \text{MaxPHYSubPacketSize}$, where MaxPHYSubPacketSize is a PHY protocol constant, then the packet size shall be N_2 and each transmission of the MAC packet will be over two PHY frames. Otherwise, two F-DCH packets of sizes N_0 and N_1 shall be generated, encoded separately, and transmitted over even-indexed and odd-indexed PHY frames, respectively.

8.6.6 Header and trailer formats

8.6.6.1 Header and trailer for transmissions not using MACID_{PAGE}

For packets not using MACID_{PAGE}, the Forward Traffic Channel shall use the following header and trailer.

8.6.6.1.1 Trailer

This protocol does not add a trailer.

8.6.6.1.2 Header

This protocol does not add a header.

8.6.6.2 Header and trailer for transmissions using MACID_{PAGE}

For packets using MACID_{PAGE}, the Forward Traffic Channel shall use the header shown in 8.6.6.2.1 and Table 105 and the trailer shown in 8.6.6.2.2.

8.6.6.2.1 Header**Table 105—MAC header format for Forward Traffic Channel**

Field	Length (bits)
PagingIDLengthType	2
ExtraPagingInformationIncluded	1
BaseAccessSequenceIncluded	1 ^a
AccessPersistenceClassIncluded	1 ^a
BaseAccessSequence	4 ^a
NumPagingIDs	5
NumPagingIDs number of the following fields:	
PagingID	32, 24, 20, or 16
AccessPersistenceClass	3 ^a
Padding	0–7

^aThis field may be absent (see field description for conditions).

PagingIDLengthType The access network shall set this field to 0 to indicate a 32 bit PagingID, to 1 to indicate a 24 bit PagingID, to 2 to indicate a 20 bit PagingID, and to 3 to indicate a 16 bit PagingID.

ExtraPagingInformationIncluded This field shall determine if the BaseAccessSequenceIncluded and AccessPersistenceClassIncluded fields are included.

BaseAccessSequenceIncluded This field shall be included if ExtraPagingInformationIncluded is equal to 1. This field shall be omitted otherwise.

AccessPersistenceClassIncluded This field shall be included if ExtraPagingInformationIncluded is equal to 1. This field shall be omitted otherwise.

BaseAccessSequence This field shall be included if the BaseAccessSequenceIncluded field is included, and equal to 1. This field shall be omitted otherwise. This field specifies the access sequence that the access terminal should use to respond to the received page.

NumPagingIDs This field shall be set to the number of PagingIDs included in the header.

PagingID Access Terminal Identifier Record. The access network shall set this field to the access terminal's PagingID, or to a number of the least significant bits of the PagingID, depending on the PagingIDLengthType field.

AccessPersistenceClass This field shall be included if **AccessPersistenceClassIncluded** field is included, and equal to 1. This field shall be omitted otherwise. This field shall be set to highest permitted persistence class that the access terminal that matches the preceding **PagingID** can use to access in response to the page.

Padding The length of this field shall be such that the entire message is octet-aligned. The sender shall set this field to zero. The receiver shall ignore this field.

8.6.6.2.2 Trailer

This protocol does not add a trailer.

8.6.7 Message formats

8.6.7.1 GroupAssignment message

The access network sends the **GroupAssignment** message to manage the access terminal's group resource allocation assignments (see Table 106).

Table 106—Group Assignment message format

Field	Length (bits)
MessageID	8
MessageSequence	8
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
AssignmentType	2
NumGroupDefinitions	3

NumGroupDefinitions occurrences of the following record (starting from GroupIndex and ending at BitmapModulation, inclusive):
{0

GroupIndex	6
GroupSignature	8
ResourceSize	2
NumChannelID	3

NumChannelID occurrences of the following fields:

ChannelID	8
SuperframeIndex	0 or 8
NumOtherGroupAllocationsRequiredToRead	2

NumOtherGroupAllocationsRequiredToRead occurrences of the following fields:

OtherGroupIndex	6
GroupInterlaceStructure	2 ^a

Table 106—Group Assignment message format (continued)

Field	Length (bits)
NumGroupInterlaceOffsets	2 ^a
GroupInterlaceOffsets	4 ^a
GroupInterlace	3 ^a
TimeChannelAllocationOrder	1 ^a
ReservedResourcesFieldUsed	1 ^a
ReservedResourcesFieldLength	3 ^a
ATPresenceAndResourceAvailabilityBitmapLength	10 ^a
ResourceAvailabilityBitmapLengthPerInterlaceOffset	1 ^a

If ResourceAvailabilityBitmapLengthPerInterlaceOffset is set to '0' or if GroupIndex is set to '000001', then one occurrence of the following record, otherwise NumGroupInterlaceOffsets occurrences of the following record (starting from ResourceAvailabilityBitmapUsed and ending at ResourceAvailabilityBitmapLength, inclusive):

{1

ResourceAvailabilityBitmapUsed	1 ^a
ResourceAvailabilityBitmapLength	7 ^a

}1

AllocationSizesOrPacketFormatsBitmapUsed	1 ^a
AllocationSizesOrPacketFormatsBitmapLength	10 ^a
AllocationSizesOrPacketFormatsBitsPerActiveAT	2 ^a
AllocationSizesOrPacketFormatsBitmapInterpretation	2 ^a
MiniMACIndexBitsUsed	1 ^a
NumMiniMACIndexBits	2 ^a
MiniMACDataPacketFormatBitsUsed	1 ^a
NumMiniMACDataPacketFormatBits	2 ^a
NumGroupPacketFormats	4 ^a

NumGroupPacketFormats+1 occurrences of the following record (starting from PacketFormat and ending at PacketFormatsBitmapEncoding, inclusive):

{1

PacketFormat	4 ^a
SpectralEfficiencyOffset	2 ^a
ModulationOffset	1 ^a
PacketFormatsBitmapEncoding	48

}1

MaxHARQTransmissions	2 ^a

Table 106—Group Assignment message format (continued)

Field	Length (bits)
BitmapChannelization	1
NumBitmapChannelIDs	2 ^a

If BitmapChannelization is equal to 0, then NumBitmapChannelIDs occurrences of the following record, otherwise one occurrence of the following record (starting from BitmapChannelID and ending at BitmapFrames, inclusive):
{1

BitmapChannelID	8
BitmapFrames	2 ^a

}1

BitmapModulation	2 ^a

}0

<i>Access Terminal Assignments</i>	
SwitchFromGroupIndex	6 ^a
AssignedGroupIndex	6 ^a
LastGroupSignature	8 ^a
MiniMACIndexAssignment	1 ^a
MiniMACIndex	4 ^a

If MiniMACIndexAssignment is set to '0' or if NumMiniMACDataPacketFormatBits for the AssignedGroupIndex is set to '11', then no occurrences of the following field. If MiniMACIndexAssignment is set to '1' and MiniMACDataPacketFormatBitsUsed for the AssignedGroupIndex is set to '0', then one occurrence of the following field. Otherwise, 2^{NumMiniMACDataPacketFormatBits} for the AssignedGroupIndex occurrences of the following field:

MiniMACDataPacketFormat	4 ^a
MiniMACMaxResources	3 ^a
NumConcurrentAssignments	2 ^a

NumConcurrentAssignments occurrences of the following record (Starting from ExtensionAssignment and ending with SharedPosition, inclusive):
{0

ExtensionAssignment	1 ^a
FirstTransmissionInterlaceOffset	3 ^a
PersistentConfiguration	2 ^a
BackToBack	1 ^a
PositionIndexPerInterlaceOffsetEnabled	1 ^a
NumUniquePositionIndices	3 ^a

Table 106—Group Assignment message format (continued)

Field	Length (bits)
If PositionIndexPerInterlaceOffsetEnabled is set to one, then NumUniquePositionIndices occurrences of the following record, otherwise one occurrence of the following record, (starting from PositionIndex and ending with SharedPosition, inclusive): {1	
PositionIndex	10 ^a
AssignedInterlaceOffsets	4 ^a
SharedPosition	1 ^a
}1	
}0	
Padding	Variable

^aThis field may be absent (see field description for conditions).

MessageID	The access network shall set this field to 0x00.
MessageSequence	For unicast messages, the access network shall set this field to 1 higher than the MessageSequence field of the last GroupAssignment message (modulo 2^S , $S = 8$) sent to this access terminal. For broadcast messages, the access network shall set this field to 1 higher than the MessageSequence field of the last GroupAssignment message (modulo 2^S , $S = 8$) sent using the broadcast MACID.
PilotID	The access network shall set this field to the PilotID corresponding to the sector for which this message is valid.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the pilot to which this message refers. This field shall be omitted otherwise.
AssignmentType	The access network shall set this field to the type of assignment that this message conveys. The encoding of this field is specified in Table 107. If this field is set to '11', this message is a broadcast message. Otherwise, this message is a unicast message.

Table 107—Assignment type

Field value (binary)	Assignment type
'00'	This assignment is a new assignment. The access terminal is assigned to the group indicated in the AssignedGroupIndex field.
'01'	This assignment moves the access terminal from the group indicated in the SwitchFromGroupIndex field to the group indicated in the AssignedGroupIndex field.
'10'	This assignment de-assigns the access terminal from the group indicated in the AssignedGroupIndex field.
'11'	This assignment conveys group properties only and can be used to change the properties of the group.

NumGroupDefinitions	The access network shall set this field to the number of group definitions included in this message.
GroupIndex	The access network shall set this field to the GroupIndex associated with the group. If the GroupIndex is set to '000001', then this group is a special group without an access terminal presence bitmap and with a reduced set of parameters.
GroupSignature	The access network shall change this field if the properties for the group corresponding to the GroupIndex field change.
ResourceSize	The access network shall set this field to one less than the fundamental resource size in units of base nodes (1 base node over 1 PHY frame).
NumChannelID	The access network shall set this field to the number of channel identifiers assigned to this group.
ChannelID	The access network shall set this field to the channel identifier assigned to this group.
SuperframeIndex	The access network shall include this field if AssignmentType is set to '11' and if the GroupIndex field is not set to '000001'. If included, the access network shall set this field to the eight least significant bits of the superframe index during which the group parameters listed below go into effect. This superframe index is an integer value s such that: $s = \lfloor t / x \rfloor$, where t represents System Time in seconds and x represents the time for a superframe in seconds as defined by the Physical Layer Protocol.
NumOtherGroupAllocationsRequiredToRead	The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to the number of other groups for which the access terminal must decode the group resource allocation bitmap in order to determine its allocation.

- OtherGroupIndex** The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to the GroupIndex of the group for which the access terminal must decode the group resource allocation bitmap in order to determine its allocation. If the NumOtherGroupAllocationsRequiredToRead field is set to '10' or '11', the access network shall list the OtherGroupIndex in the order in which the groups are allocated resources.
- GroupInterlaceStructure** The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to the group interlace structure of the time domain resources, which are assigned to the group. The encoding of this field is as specified in Table 108.

Table 108—Group interlace structure encoding for FDD

Field value (binary)	Interlace duration (frames), number of consecutive frames
'00'	FL 8-interlace structure without extended transmission.
'01'	FL 8-interlace structure with two PHY frame transmission.
'10'	FL 6-interlace structure without extended transmission.
'11'	FL 6-interlace structure with two PHY frame.

Table 109—Group Interlacing Structure Encoding for TDD4:4

Field value (binary)	Interlace duration (frames), number of consecutive frames
'00'	FL TDD 4:4 4-interlace structure.
'01'	FL TDD 4:4 4-interlace structure with two PHY frames transmissions.
'10'	Reserved.
'11'	Reserved.

Table 110—Group Interlacing Structure Encoding for TDD6:3

Field value (binary)	Interlace duration (frames), number of consecutive frames
'00'	FL TDD 6:3 3-interlace structure.
'00'	Reserved.
'10'	Reserved.
'11'	Reserved.

NumGroupInterlaceOffsets

The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to one less than the number of group interlace offsets, which are defined for the group.

GroupInterlaceOffsets	The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to indicate the group interlace offsets in which this group is valid. A '1' in the $(k - 1)^{\text{th}}$ position indicates that the group parameters contained in this message for GroupIndex are valid during group interlace offset $k - 1$. A '0' in the $(k - 1)^{\text{th}}$ position of this field indicates that the group parameters contained in this message for GroupIndex are not valid during group interlace offset $k - 1$. The least significant bit corresponds to the 0^{th} position, which further corresponds to group interlace offset 0.
GroupInterlace	The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to the group interlace assigned to the group. The least significant bit corresponds to the 0^{th} position, which further corresponds to group interlace offset 0.
ResourceSize	The access network shall set this field to one less than the fundamental resource size in units of base-nodes (one base-node over one PHY frame).
NumChannelIID	The access network shall set this field to the number of channel identifiers assigned to this group.
ChannelID	The access network shall set this field to the channel identifier assigned to this group.
TimeChannelAllocationOrder	The access network shall omit this field if GroupIndex is set to '000001'. The access network shall set this field to '0' if resources are allocated in ascending order beginning with frames followed by channels, with the first resource allocated being the first channel in the first frame, and shall set this field to '1' if resources are allocated in descending order beginning with frames followed by channels, with the first resource allocated being the last channel in the last frame.
ReservedResourcesFieldUsed	The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to '1' if the reserved resources field is used and to '0' otherwise.
ReservedResourcesFieldLength	The access network shall include this field if the ReservedResourcesFieldUsed field is set to '1'. If included, the access network shall set this field to one less than the length of the reserved resources field in units of bits.
ATPresenceAndResourceAvailabilityBitmapLength	The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to one less than the length of the resource availability bitmap plus the length of the access terminal presence bitmap in units of bits.
ResourceAvailabilityBitmapLengthPerInterlaceOffset	The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to '1' to indicate that the length of the resource availability bitmap is made per group interlace offset and shall set this field to '0' to indicate that the length of the resource availability bitmap is the same for all group interlace offsets.

ResourceAvailabilityBitmapUsed

The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to '1' if the resource availability bitmap is used and to '0' otherwise.

ResourceAvailabilityBitmapLength

The access network shall include this field if the ResourceAvailabilityBitmapUsed field is set to '1' or if the GroupIndex is set to '000001'. If included, the access network shall set this field to one less than the length of the resource availability bitmap in units of bits. Each bit in the resource availability bitmap represents one resource, where a resource is defined as a series of N sequential base-nodes (sequential in time and frequency according to the numbering) within the group resources, where N is given by the ResourceSize field.

AllocationSizesOrPacketFormatsBitmapUsed

The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to '1' if the allocation sizes or packet formats bitmap is used and '0' otherwise.

AllocationSizesOrPacketFormatsBitmapLength

The access network shall include this field only if the AllocationSizesOrPacketFormatsBitmapUsed field is set to '1' and if the GroupIndex field is not set to '000001'. If included, the access network shall set this field to the one less than length of the allocation sizes or packet formats bitmap in units of bits.

AllocationSizesOrPacketFormatsBitsPerActiveAT

The access network shall include this field only if the AllocationSizesOrPacketFormatsBitmapUsed field is set to '1' and if the GroupIndex field is not set to '000001'. If included, the access network shall set this field to one less than the number of bits contained in the allocation sizes or packet formats bitmap per active access terminal.

AllocationSizesOrPacketFormatsBitmapInterpretation

The access network shall include this field only if the AllocationSizesOrPacketFormatsBitmapUsed field is set to '1' and if the GroupIndex field is not set to '000001'. If included, the access network shall set this field to indicate how the access terminal is to interpret the allocation sizes or packet formats bitmap. The encoding of this field is specified in Table 111.

Table 111—Allocation sizes or packet formats bitmap interpretation

Field value (binary)	Allocation sizes or packet formats bitmap interpretation
'00'	An allocation sizes bitmap is used. The number of allocated resources is one more than the binary value indicated in the allocation sizes bitmap for the access terminal. A single packet is encoded and transmitted on the number of allocated resources.

Table 111—Allocation sizes or packet formats bitmap interpretation (continued)

Field value (binary)	Allocation sizes or packet formats bitmap interpretation
'01'	An allocation sizes bitmap is used. The number of allocated resources is one more than the binary value indicated in the allocation sizes bitmap for the access terminal. If more than one resource is allocated: (1) Two individually encoded packets (P1 and P2) are transmitted on evenly divided portions of NAR, where NAR is the number of allocated resources. P2 is transmitted on the first NAR/2 resources within the access terminal's allocation, and P1 is transmitted on the last NAR/2 resources within the access terminal's allocation. (2) The HARQ transmission opportunity for P2 is less than the HARQ transmission opportunity for P1 by the number of group interlace offsets.
'10'	The packet formats bitmap is used. The packet format used for the access terminal is indicated in the packet formats bitmap. The value in the packet formats bitmap corresponds to the group packet format in the list of GroupPacketFormats.
'11'	Reserved.

MiniMACIndexBitsUsed The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to '1' if the group resource allocation bitmap contains a Mini MAC field and '0' otherwise.

NumMiniMACIndexBits The access network shall include this field only if the MiniMACIndexBitsUsed field is set to '1' and if the GroupIndex field is not set to '000001'. If included, the access network shall set this field to one less than the number of bits used to convey the MiniMACIndex.

MiniMACDataPacketFormatBitsUsed
The access network shall include this field only if the MiniMACIndexBitsUsed field is set to '1' and if the GroupIndex field is not set to '000001'. If included, the access network shall set this field to '1' if packet format bits are included in the mini MAC field, and '0' otherwise.

NumMiniMACDataPacketFormatBits
The access network shall include this field only if the MiniMACDataPacketFormatBitsUsed field is set to '1' and if the GroupIndex field is not set to '000001'. If included, the access network shall set this field to one less than the number of bits used to convey the packet format for the access terminal corresponding to the mini MAC Index.

NumGroupPacketFormats
The access network shall omit this field if GroupIndex is set to '000001'. The access network shall set this field to one less than the number of packet data formats that shall be supported by the access terminals indicated as active in the resource availability bitmap or the access terminal presence bitmap.

PacketFormat
The access network shall include this field if NumGroupPacketFormats is not set to '1111' and if the GroupIndex field is not set to '000001'. If included, the access network shall set this field to the index of the packet format.

SpectralEfficiencyOffset The access network shall include this field if the GroupIndex field is not set to '000001' and if the PacketFormat field is set to '0010', '0011', or '0100'. If included, the access network shall set this field to the spectral efficiency offset

that is to be applied to the spectral efficiency associated with the packet format indicated by the PacketFormat field. The modified spectral efficiency is defined as the spectral efficiency associated with the packet format indicated by the PacketFormat field plus the value indicated by this field. The encoding of this field is specified in Table 112.

Table 112—Spectral efficiency offset encoding

Field value (binary)	Spectral efficiency offset
'00'	0.000
'01'	−0.125
'10'	−0.250
'11'	−0.375

ModulationOffset

The access network shall include this field if the GroupIndex field is not set to '000001' and if the PacketFormat field is set to '0011' or '0100'. If included, the access network shall set this field to '0' if the access terminal is to use the modulation orders associated with the packet format indicated by the PacketFormat field, and shall set this field to '1' if the access terminal is to use QPSK modulation for all HARQ transmissions for the packet format indicated by the PacketFormat field.

PacketFormatsBitmapEncoding

The access network shall include this field only if AllocationSizesOrPacketFormatsBitmapUsed is set to '1' and AllocationSizesOrPacketFormatsBitmapInterpretation is set to '10' and GroupIndex is not set to '000001'. If included, the access network shall set the least significant bits of this field to the value that will be included in the packet formats bitmap if this group packet format is used.

MaxHARQTransmissions

The access network shall omit this field if GroupIndex is set to '000001'. If included, the access network shall set this field to the maximum number of HARQ transmissions. The encoding of this field is as specified in Table 113.

Table 113—Maximum HARQ transmissions encoding

Field value (binary)	Maximum HARQ transmissions
'00'	3
'01'	4
'10'	5
'11'	6

BitmapChannelization

The access network shall set this field to '0' if the channel identifiers listed in the following correspond to NodeIDs from the channel tree and shall set this field to '1' if the channel identifier listed below corresponds to LAB index.

- NumBitmapChannelIDs** The access network shall include this field if **BitmapChannelization** is equal to '0'. If included, the access network shall set this field to the number of channel identifiers that are used to transmit the bitmap.
- BitmapChannelID** The access network shall set this field to the channel identifier assigned for transmitting the bitmap.
- BitmapFrames** The access network shall include this field if **GroupInterlaceStructure** is included and equal to '01' or '11'. If included, the access network shall set this field as specified in Table 114.

Table 114—Bitmap Frames Encoding

Field value (binary)	Bitmap frames
'00'	Reserved.
'01'	Bitmap Occupies Channel Only in Second Frame.
'10'	Bitmap Occupies Channel Only in First Frame.
'11'	Bitmap Occupies Channel in Both Frames.

- BitmapModulation** The access network shall omit this field if **GroupIndex** is set to '000001'. If included, the access network shall set this field to the modulation used for the group resource allocation bitmap. The encoding of this field is as specified in Table 115.

Table 115—Bitmap Modulation Encoding

Field value (binary)	Bitmap modulation
'00'	QPSK
'01'	8-PSK
'10'	16-QAM
'11'	64-QAM

The access network shall omit the remaining fields if **GroupIndex** is set to '000001' and **NumGroupDefinitions** is set to '001' or if **AssignmentType** is set to '11'.

- SwitchFromGroupIndex** The access network shall include this field only if **AssignmentType** is set to '01'. If included, the access network shall set this field to the **GroupIndex** associated with the group from which the access terminal is to switch.
- AssignedGroupIndex** If the **AssignmentType** field is '00' or '01', the access network shall set this field to the **GroupIndex** field of the group to which the access terminal is assigned. If the **AssignmentType** field is '10', the access network shall set this field to the **GroupIndex** field of the group from which the access terminal is deassigned.
- LastGroupSignature** The access network shall include this field if the group properties for the **AssignedGroupIndex** are not included in this message and if the

AssignmentType field is set to '00' or '01'. If included, the access network shall set this field to the value of the last GroupSignature field that was used to define or change the group properties for the AssignedGroupIndex.

MiniMACIndexAssignment

The access network shall include this field only if the AssignmentType field is set to '00' or '01'. If included, the access network shall set this field to '1' if the access terminal is to monitor the MiniMACIndex bits of the group resource allocation bitmap for determining a resource allocation and '0' otherwise.

MiniMACIndex

The access network shall include this field only if the MiniMACIndexAssignment field is set to '1'. If included, the access network shall set this field to the MiniMACIndex assigned to the access terminal.

MiniMACDataPacketFormat

The access network shall include this field only if the MiniMACIndexAssignment is set to '1' and if the NumMiniMACDataPacketFormatBits field for group corresponding to the AssignedGroupIndex is not set to '11'. If included, the access network shall set this field to the index of the packet format as defined in the physical layer protocol. The Nth PacketFormat in the list shall be indicated by setting the value in the packet format bits to the binary equivalent.

MiniMACMaxResources

The access network shall include this field only if the MiniMACIndexAssignment field is set to '1'. If included, the access network shall set this field to the maximum number of resources that will be used for the access terminal, irrespective of the actual number of unused resources. The encoding of this field is specified in Table 116.

Table 116—Mini MAC max resources

Field value (binary)	Mini MAC max resources
'000'	4
'001'	8
'010'	16
'011'	32
'100'	64
'101'	96
'110'	128
'111'	Unlimited

NumConcurrentAssignments

The access network shall include this field only if AssignmentType is set to '00' or '01'. If included, the access network shall set this field to the number of concurrent assignments for the access terminal addressed in this message. The encoding of this field is specified in Table 117.

Table 117—Number of concurrent assignments

Field value (binary)	Number of concurrent assignments
'00'	0
'01'	1
'10'	2
'11'	Reserved

ExtensionAssignment The access network shall include this field if NumConcurrentAssignments is set to '10'. If included, the access network shall set this field to '1' if the associated set of position indices are used as a HARQ extension for the set of position indices indicated when this field was set to '0' and shall set this field to '0' if the associated set of position indices are not used as a HARQ extension.

FirstTransmissionInterlaceOffset The access network shall omit this field if NumConcurrentAssignments is set to '10' and ExtensionAssignment is equal to '1'. If included, the access network shall set this field to the group interlace offset for which a first HARQ transmission opportunity is defined.

PersistentConfiguration The access network shall set this field to indicate the persistent configuration for the access terminal. The encoding of this field is as specified in Table 118.

Table 118—Persistent configuration

Field value (binary)	Persistent configuration
'00'	The position assignment corresponding to the first transmission group interlace offset shall correspond to one of the bits in the resource availability bitmap; i.e., the resource is persistently allocated and the access terminal does not need to decode the group resource allocation bitmap for the first transmission group interlace offset. The position assignment(s) corresponding to other group interlace offsets shall correspond to one of the bits in the access terminal presence bitmap; i.e., the resources are not persistently allocated and the access terminal shall decode the group resource allocation bitmap to determine the resource allocated on those group interlace offsets.
'01'	The position assignment corresponding to each of the group interlace offsets shall correspond to one of the bits in the resource availability bitmap; i.e., the resources are persistently allocated and the access terminal does not need to decode the group resource allocation bitmap for any group interlace offset.
'10'	The position assignment corresponding to the first transmission group interlace offset shall correspond to one of the bits in the access terminal presence bitmap. The resource allocated shall persist for subsequent HARQ transmission opportunities of the same packet. The access terminal does not need to decode the group resource allocation bitmap for group interlace offsets that are not corresponding to the first transmission group interlace offset.
'11'	The position assignment corresponding to each of the group interlace offsets shall correspond to one of the bits in the access terminal presence bitmap.

BackToBack The access network shall include this field if PersistentConfiguration is set to '10'. If included, the access network shall set this field to '1' if the access network may transmit back-to-back packets to the access terminal, irrespective of the group interlace offset, and the access network shall set this field to '0' if the access network shall not transmit back-to-back packets to the access terminal without transmitting a F-SPCH, with an 'ON' state, corresponding to the base node with the lowest index in the earliest frame within access terminal's allocation.

PositionIndexPerInterlaceOffsetEnabled The access network shall omit this field if PersistentConfiguration is set to '10'. If included, the access network shall set this field '1' if the PositionIndex assignment is made per group interlace offset and shall set this field to '0' to indicate that the PositionIndex assignment is valid for all group interlace offsets for this group.

NumUniquePositionIndices This access network shall include this field only if PositionIndexPerInterlaceOffsetEnabled is included and is set to '1'. If included, the access network shall set this field to one less than the number of unique PositionIndex fields that are assigned to the access terminal.

PositionIndex The access network shall set this field to the position assigned to the access terminal within the group. The N^{th} group position is associated with the N^{th} position in the concatenation of the resource availability bitmap and the access terminal presence bitmap.

AssignedInterlaceOffsets This field is present only if PositionIndexPerInterlaceOffsetEnabled is included and is set to '1'. If included, the access network shall set this field to indicate the group interlace offsets associated with the assigned PositionIndex field above. A '1' in the k^{th} position indicates that the corresponding PositionIndex (specified in the previous field) is valid during group interlace offset $k-1$. A '0' in the k^{th} position of this field indicates that the access terminal will not be indicated in the group resource allocation bitmap during group interlace offset $k-1$ with the PositionIndex specified in the previous field.

SharedPosition The access network shall set this field to '1' if this position assignment is shared with other access terminals, and shall set this field to '0' otherwise.

Padding The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros.

Channels	FTC	FTC
Addressing	Broadcast	Unicast

RLP	Reliable	Best effort
AuthTag	Not Required	

This message is sent reliably when the AssignmentType field is set to '00', '01', or '10' and is sent in a best effort manner when AssignmentType field is set '11'.

8.6.7.2 GroupAssignmentComplete

If the AssignmentType field of the GroupAssignment message is not set to '11', the access terminal shall send the GroupAssignmentComplete message to provide an acknowledgment for the GroupAssignment message (see Table 119).

Table 119—GroupAssignmentComplete message

Field	Length (bits)
MessageID	8
MessageSequence	8

MessageID The access terminal shall set this field to 0x01.

MessageSequence The access terminal shall set this field to the MessageSequence field of the GroupAssignment message whose receipt this message is acknowledging.

Channels	RTC
Addressing	Unicast

RLP	Best effort
AuthTag	Required if key is available

8.6.7.3 GroupPropertiesRequest

The access terminal shall send the GroupPropertiesRequest message to the access network if it receives a GroupAssignment message assigning it to a group for which it does not have the current set of group properties. (See Table 120.)

Table 120—GroupPropertiesRequest message format

Field	Length (bits)
MessageID	8
MessageSequence	8
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
GroupIndex	6
Reserved	7

MessageID The access terminal shall set this field to 0x02.

MessageSequence The access terminal shall set this field to 1 higher than the MessageSequence field of the last GroupPropertiesRequest message (modulo 2^S , $S = 8$) sent to the access network.

PilotID The access terminal shall set this field to PilotID of the sector for which it is requesting group parameters.

ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.
GroupIndex	The access terminal shall set this field to the GroupIndex for which it is requesting group parameters.
Reserved	The receiver shall ignore this field.

Channels	RTC

RLP	Best effort

Addressing	Unicast

AuthTag	Required if key is available

8.6.7.4 CodebookAssignment

The access network sends this message to provide a Codebook to the access terminal. This message supports fragmentation of codebooks, and allows the access network to transmit only part of a Codebook in this message. If the access network performs fragmentation, a Codebook is usable at the access terminal only when it receives all matrices and the OverlapMap field.

The access terminal shall maintain a cache of an implementation-dependent number of received Codebooks, and the access terminal may delete cache entries in an implementation dependent manner. (See Table 121.)

Table 121—Codebook Assignment message format

Field	Length (bits)
MessageID	8
CodebookID	16
NumClusters	8
NumMatrices	8
FirstMatrixInMessage	8
NumMatricesInMessage	8
NumMatricesInMessage of the following record{	
NumAntennas	4
SpatialOrder	4
ClusterID	8
NumAntennas x SpatialOrder of the following record{	
MatrixEntryReal	8
MatrixEntryImag	8
}	

Table 121—Codebook Assignment message format (continued)

Field	Length (bits)
OverlapMap	$\text{NumClusters} \times (\text{NumClusters} - 1) / 2$
Padding	Variable

MessageID	The access network shall set this field to 0x03.
CodebookID	The access network shall set this field to CodebookID of the codebook contained in this message.
NumClusters	The number of clusters in this message. This field shall take values between 1 and NumMatrices (both inclusive).
NumMatrices	The number of matrices in the codebook. Matrices with indices 1 through NumMatrices (both inclusive) are communicated to the access terminal as part of the Codebook. No matrix is included for index 0, and index 0 is used by the access terminal to indicate no preferred matrix. This field shall take values between 1 and 63 (both inclusive).
FirstMatrixInMessage	The index of the first matrix included in this message. This field shall take values from 1 to NumMatrices (both inclusive).
NumMatricesInMessage	The number of matrices included in this message. This field shall take values from 1 to NumMatrices (both inclusive).
NumAntennas	The number of antennas (rows) for this matrix. This field shall take values between 1 and 8 (both inclusive).
SpatialOrder	The number of spatial dimensions (columns) for this matrix. This field shall take values between 1 and 4 (both inclusive).
ClusterID	The access network shall set this field to the ClusterID of the code matrix described next.
MatrixEntryReal	Related to the real part of the matrix entry, listed row first. The matrix entry shall take the value $(128 - 2^n)/128$, where n is the value of the eight-bit field.
MatrixEntryImag	Related to the imaginary part of the matrix entry, listed row first. The matrix entry shall take the value $(128 - 2^n)/128$, where n is the value of the eight bit field.
OverlapMap	This field shall be included if NumMatrices is equal to NumMatricesInMessage + FirstMatrixInMessage (i.e., if this message contains the last matrix of the codebook). The $(\text{NumClusters} \times i + j)$ entry of this bit field shall be set to 1 if cluster i and cluster j overlap, and shall be set to zero otherwise. Here i and j take values from 0 to NumClusters – 1.

Padding The length of this field shall be such that this message is octet aligned. The sender shall set this field to zero. The receiver shall ignore this field.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

8.6.7.5 CodebookQuery

The access network sends this message to query the Codebooks that the access terminal currently has in its cache (see Table 122).

Table 122—Codebook Query Message Format

Field	Length (bits)
MessageID	8

MessageID The access network shall set this field to 0x04.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

8.6.7.6 CodebookResponse

The access terminal shall send this message in response to a CodebookQuery message. The access terminal shall include the CodebookIDs for all Codebooks it has in its cache (see Table 123).

Table 123—CodebookResponse message format 1039

Field	Length (bits)
MessageID	8
NumCodebookIDs	8
NumCodebookIDs occurrences of the following field	
CodebookID	16

MessageID The access terminal shall set this field to 0x05.

NumCodebookIDs The access terminal shall set this field to the number of CodebookIDs included in this message.

CodebookID The access terminal shall set this field to a CodebookID for a usable Codebook that it has in its codebook cache. The access terminal should include all entries in its Codebook cache in this message.

Channels	RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

8.6.8 Interface to other protocols

8.6.8.1 Commands sent

- *ReverseControlChannelMAC.ChangeFLSS*
- *ReverseControlChannelMAC.ACKAssignment*

8.6.8.2 Indications

This protocol registers to receive the following indications:

- *AirLinkManagement.TunedAway*
- *ReverseControlChannelMAC.FLSSChanged*

8.6.9 Configuration attributes

8.6.9.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 124. The access terminal shall use as defaults the values in Table 124 that are listed in ***bold italics***.

Table 124—Configurable simple attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	ROFDMSupportedFL Capability: access network shall not propose this attribute	Soft Allowed/Static	<i>0x00</i>	The access terminal does not support Rotational OFDM on FL. The access network shall not set the ROFDMEnabledFL field in the MACResourceAssignment message of the Active Set Management Protocol to '1'.
			0x01	The access terminal supports Rotational OFDM on FL.
			All other values	Reserved.
0x0001	MaxSCWStreams Capability: access network shall not propose this attribute	Soft Allowed/Static	<i>0x01</i>	The maximum number of SCW streams supported by the access terminal.
			0x02–0x04	The maximum number of SCW streams supported by the access terminal.
			All other values	Reserved.
0x0002	MaxMCWLayers Capability: access network shall not propose this attribute	Soft Allowed/Static	<i>0x00</i>	MCW MIMO is not supported by the access terminal.
			0x01–0x04	The maximum number of MCW Layers supported by the access terminal.
			All other values	Reserved

Table 124—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0003	MaxSTTDRank Capability: access network shall not propose this attribute	Soft Allowed/Static	0x00	The access terminal does not support STTD with any Rank. The value 0x00 is not allowed if the access terminal supports a maximum number of SCW streams equal to or larger than 2; or a maximum number of MCW layers equal to or larger than 2.
			0x01	The access terminal supports STTD rank 1. Support for STTD rank 1 is mandatory if the access terminal supports a maximum number of SCW streams equal to or larger than 2; or a maximum number of MCW layers equal to or larger than 2.
			0x02	The access terminal supports STTD rank 1 and 2. Support for STTD rank 1 and rank 2 is mandatory if the access terminal supports a maximum number of SCW streams of equal or larger than 4; or a maximum number of MCW layers of equal or larger than 4; or a maximum number of precoded transmit antennas of equal or larger than 4.
			All other values	Reserved.
0x0004	STTDAntennaSelectionSupported Capability: access network shall not propose this attribute	Soft Allowed/Static	0x00	The access terminal does not support STTDAntennaSelection.
			0x01	The access terminal supports STTDAntennaSelection.
			All other values	Reserved.
0x0005	MaxPrecodedTxAntennas Capability: access network shall not propose this attribute	Soft Allowed/Static	0x02	The maximum number of precoded transmit antennas supported by the access terminal.
			0x03–0x08	The maximum number of precoded transmit antennas supported by the access terminal.
			All other values	Reserved.

Table 124—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0006	MaxNumFLSDMASubtrees Capability: access network shall not propose this attribute	Soft Allowed/Static	0x02	The maximum number of SDMA subtrees supported by the access terminal on FL.
			0x03–0x08	The maximum number of SDMA subtrees supported by the access terminal on FL.
			All other values	Reserved.
0x0007	MaxGRAGroups	Soft Allowed/Static	0x02	The number of GRA Groups supported by the access terminal.
			0x03–0x06	The number of GRA Groups supported by the access terminal.
			All other values	Reserved.
0x0008	EnhancedPageFeaturesSupport	Soft Allowed/Static	0x00	The access terminal does not support enhanced page features.
			0x01	The access terminal supports enhanced page features.
			All other values	Reserved.
0x0009	SupercastSupported Capability: access network shall not propose this attribute	Soft Allowed/Static	0x00	The access terminal does not support the reception of Supercast.
			0x01	The access terminal supports the reception of Supercast.
			All other values	Reserved.
0x000a	FTCSupervisionTimeout	Soft Allowed/Dynamic	0x03	The CQI supervision timeout value in seconds.
			0x02 and 0x04–0x0a	The CQI supervision timeout value in seconds.
			All other values	Reserved.
0x000b	PilotFormat2Supported Capability: Access network shall not propose values for this attribute	Soft Allowed/Dynamic	0x00	The access terminal does not support pilot format 2 on F-DCH.
			All other values	Reserved.

8.6.9.2 Complex attributes

8.6.9.2.1 FLMultiHypothesisDecodeProfileK

This attribute allows the access network to send and the access terminal to receive more than one packet format for an assignment that contains a given packet format. The access network shall not use a given profile for transmissions to an access terminal if the access terminal does not support that profile. (See Table 125.)

This attribute has AttributeID 0x810K.

Table 125—Definition of FLMultiHypothesisDecodeProfileK

Field	Length (bits)	Default
FLMultiDecodeAssignedPFK	4	0
FLMultiDecodeAssignmentSizeK	8	0
FLMultiDecodeNumCompatiblePFK	4	0
FLMultiDecodeNumCompatiblePFK occurrences of the following field:		
FLMultiDecodeCompatiblePFKj	4	N/A
}		
Padding	Variable	N/A

FLMultiDecodeAssignedPFK

For the multiple hypothesis profile K , this field shall determine the packet format for which the access terminal tries multiple packet format hypothesis.

FLMultiDecodeAssignmentSizeK

For the multiple hypothesis profile K , this field shall determine the assignment size, in units of BaseNodes, for which the access terminal tries multiple packet format hypothesis.

FLMultiDecodeNumCompatiblePFK

For the multiple hypothesis profile K , this field shall be set to the number of packet formats, in addition to FLMultiDecodeAssignedPF, that the access terminal is required to decode when it receives an assignment requiring multiple decode hypotheses.

FLMultiDecodeCompatiblePFKj

For the multiple hypothesis profile K , the j^{th} occurrence of this field shall be set to a packet format that the access terminal is required to decode when it receives an assignment requiring multiple decode hypotheses.

Padding

The sender shall set the size of this field such that the attribute is octet aligned. The receiver shall ignore this field.

Commit	Soft allowed

Scope	Dynamic

8.6.9.2.2 FLSupportedPacketSizeEightInterlace

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating on the forward link for packets using 8-interlace non-extended assignments using turbo coding in the FDD mode. (See Table 126.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8000.

Table 126—Definition of FLSupportedPacketSizeEightInterlace

Field	Length (bits)	Default
MaxPacketSizeEightInterlace	8	1
MaxRateOneFifthPacketSizeEightInterlace	8	1
MaxRateOneThirdPacketSizeEightInterlace	8	1
MaxRateOneHalfPacketSizeEightInterlace	8	1

MaxPacketSizeEightInterlace

This field shall be set to the maximum packet size that the access terminal can decode in the 8-interlace interlacing structure.

MaxRateOneFifthPacketSizeEightInterlace

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-fifth packet using turbo coding.

MaxRateOneThirdPacketSizeEightInterlace

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-third packet using turbo coding.

MaxRateOneHalfPacketSizeEightInterlace

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-half packet using turbo coding.

Commit	Soft allowed	Scope	Static

8.6.9.2.3 FLSupportedPacketSizeEightInterlaceExtended

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating on the forward link for packets using 8-interlace extended assignments using turbo coding in the FDD mode. (See Table 127.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8001.

Table 127—Definition of FDDFLSupportedPacketSizeEightInterlaceExtended

Field	Length (bits)	Default
MaxPacketSizeEightInterlaceExtended	8	0

MaxPacketSizeEightInterlaceExtended

This field shall be set to the maximum packet size that the access terminal can decode in the 8-interlace interlacing structure with extended frames using turbo coding.

Commit	Soft allowed

Scope	Static

8.6.9.2.4 FLSupportedPacketSizeSixInterlace

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating on the forward link for packets using 6-interlace assignments and turbo coding in the FDD mode. (See Table 128.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8002.

Table 128—Definition of FLSupportedPacketSizeSixInterlace

Field	Length (bits)	Default
MaxPacketSizeSixInterlace	8	0
MaxRateOneFifthPacketSizeSixInterlace	8	0
MaxRateOneThirdPacketSizeSixInterlace	8	0
MaxRateOneHalfPacketSizeSixInterlace	8	0

MaxPacketSizeSixInterlace

This field shall be set to the maximum packet size that the access terminal can decode in the 6-interlace interlacing structure using turbo coding.

MaxRateOneFifthPacketSizeSixInterlace

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-fifth packet using turbo coding.

MaxRateOneThirdPacketSizeSixInterlace

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-third packet using turbo coding.

MaxRateOneHalfPacketSizeSixInterlace

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-half packet using turbo coding.

Commit	Soft allowed

Scope	Static

8.6.9.2.5 FLSupportedPacketSizeEightInterlaceLDPC

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating on the forward link for packets using 8-interlace non-extended assignments and LDPC encoding, if DuplexingMode = FDD. (See Table 129.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8003.

Table 129—Definition of FLSupportedPacketSizeEightInterlaceLDPC

Field	Length (bits)	Default
MaxPacketSizeEightInterlaceLDPC	8	0
MaxRateOneFifthPacketSizeEightInterlaceLDPC	8	0
MaxRateOneThirdPacketSizeEightInterlaceLDPC	8	0
MaxRateOneHalfPacketSizeEightInterlaceLDPC	8	0

MaxPacketSizeEightInterlaceLDPC

This field shall be set to the maximum packet size that the access terminal can decode in the 8-interlace interlacing structure using LDPC coding.

MaxRateOneFifthPacketSizeEightInterlaceLDPC

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-fifth packet using LDPC coding.

MaxRateOneThirdPacketSizeEightInterlaceLDPC

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-third packet using LDPC coding.

MaxRateOneHalfPacketSizeEightInterlaceLDPC

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-half packet using LDPC coding.

Commit	Soft allowed

Scope	Static

8.6.9.2.6 FLSupportedPacketSizeSixInterlaceLDPC

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating on the forward link for packets using 6-interlace assignments and LDPC coding, if DuplexingMode = FDD. (See Table 130.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8004.

Table 130—Definition of FLSupportedPacketSizeSixInterlaceLDPC

Field	Length (bits)	Default
MaxPacketSizeSixInterlaceLDPC	8	0
MaxRateOneFifthPacketSizeSixInterlaceLDPC	8	0
MaxRateOneThirdPacketSizeSixInterlaceLDPC	8	0
MaxRateOneHalfPacketSizeSixInterlaceLDPC	8	0

MaxPacketSizeSixInterlaceLDPC

This field shall be set to the maximum packet size that the access terminal can decode in the 6-interlace interlacing structure using LDPC coding.

MaxRateOneFifthPacketSizeSixInterlaceLDPC

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-fifth packet using LDPC coding.

MaxRateOneThirdPacketSizeSixInterlaceLDPC

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-third packet using LDPC coding.

MaxRateOneHalfPacketSizeSixInterlaceLDPC

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-half packet using LDPC coding.

Commit	Soft allowed

Scope	Static

8.6.9.2.7 FLSupportedPacketSizeTDD44

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating on the forward link for packets using 4-interlace assignments without extended transmissions if DuplexingMode = TDD44. (See Table 131.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8020.

Table 131—Definition of FLSupportedPacketSizeTDD44

Field	Length (bits)	Default
MaxRateOneFifthPacketSizeTDD44	8	0
MaxRateOneThirdPacketSize TDD44	8	0
MaxRateOneHalfPacketSize TDD44	8	0

MaxRateOneFifthPacketSizeTDD44

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-fifth packet.

MaxRateOneThirdPacketSizeTDD44

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-third packet.

MaxRateOneHalfPacketSizeTDD44

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-half packet.

Commit	Soft allowed

Scope	Static

8.6.9.2.8 FLSupportedPacketSizeLDPCTDD44

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating on the forward link for packets using 4-interlace assignments and LDPC coding in TDD44. (See Table 132.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8021.

Table 132—Definition of FLSupportedPacketSizeLDPCTDD44

Field	Length (bits)	Default
MaxPacketSizeLDPCTDD44	8	0
MaxRateOneFifthPacketSizeLDPCTDD44	8	0
MaxRateOneThirdPacketSize LDPCTDD44	8	0
MaxRateOneHalfPacketSize LDPCTDD44	8	0

MaxPacketSizeLDPCTDD44

This field shall be set to the maximum packet size that the access terminal can decode in the 4-interlace interlacing structure using LDPC coding.

MaxRateOneFifthPacketSize LDPCTDD44

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-fifth packet using LDPC coding.

MaxRateOneThirdPacketSizeFourInterlace

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-third packet using LDPC coding.

MaxRateOneHalfPacketSizeFourInterlace

This field shall be set to the maximum packet size that the access terminal can decode for a rate one-half packet using LDPC coding.

Commit	Soft allowed

Scope	Static

8.6.9.2.9 FLSupportedPacketSizeTDD63

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating one FL PHY frame on the forward link for packets using three interlace assignments (without extended transmissions in) if DuplexingMode = TDD63. (See Table 133.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8022.

Table 133—Definition of FLSupportedPacketSizeTDD63

Field	Length (bits)	Default
MaxRateOneFifthPacketSizeTDD63	8	0
MaxRateOneThirdPacketSizeTDD63	8	0
MaxRateOneHalfPacketSizeTDD63	8	0

MaxRateOneFifthPacketSizeTDD63

This field shall be set to the maximum packet size that the access terminal can decode in one FL PHY frame for a rate one-fifth packet.

MaxRateOneThirdPacketSizeTDD63

This field shall be set to the maximum packet size that the access terminal can decode in one FL PHY frame for a rate one-third packet.

MaxRateOneHalfPacketSizeTDD63

This field shall be set to the maximum packet size that the access terminal can decode in one FL PHY frame for a rate one-half packet.

Commit	Soft allowed

Scope	Static

8.6.9.2.10 FLSupportedPacketSizeLDPCTDD63

This attribute describes the maximum packet size, in units of kilobits, that the access terminal is capable of demodulating in one FL PHY frame on the forward link for packets using three interlace and LDPC coding in DuplexingMode = TDD6344. (See Table 134.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8023.

Table 134—Definition of TDDFLSupportedPacketSizeEightInterlaceLDPC

Field	Length (bits)	Default
MaxPacketSizeEightInterlaceLDPCTDD63	8	0
MaxRateOneFifthPacketSizeEightInterlaceLDPC TDD63	8	0
MaxRateOneThirdPacketSizeEightInterlaceLDPC TDD63	8	0
MaxRateOneHalfPacketSizeEightInterlaceLDPC TDD63	8	0

MaxPacketSizeEightInterlaceLDPCTDD63

This field shall be set to the maximum packet size that the access terminal can decode in one FL PHY frame in the three-interlace interlacing structure using LDPC coding.

MaxRateOneFifthPacketSizeEightInterlaceLDPCTDD63

This field shall be set to the maximum packet size that the access terminal can decode in one FL PHY frame for a rate one-fifth packet using LDPC coding.

MaxRateOneThirdPacketSizeEightInterlaceLDPCTDD63

This field shall be set to the maximum packet size that the access terminal can decode in one FL PHY frame for a rate one-third packet using LDPC coding.

MaxRateOneHalfPacketSizeEightInterlaceLDPCTDD63

This field shall be set to the maximum packet size that the access terminal can decode in one FL PHY frame for a rate one-half packet using LDPC coding.

Commit	Soft allowed

Scope	Static

8.6.9.2.11 FLSupportedAssignmentSize

This attribute describes the maximum assignment size, in units of 16 subcarriers, that the access terminal is capable of demodulating on the forward link. (See Table 135.)

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8005.

Table 135—Definition of FLSupportedAssignmentSize

Field	Length (bits)	Default
MaxAssignmentSizeSixInterlaceSISO	8	0
MaxAssignmentSizeSixInterlaceSCW	8	0
MaxAssignmentSizeSixInterlaceMCW	8	0
MaxAssignmentSizeEightInterlaceSISO	8	32
MaxAssignmentSizeEightInterlaceSCW	8	0
MaxAssignmentSizeEightInterlaceMCW	8	0
MaxAssignmentSizeExtendedFrameSISO	8	0
MaxAssignmentSizeExtendedFrameSCW	8	0
MaxAssignmentSizeExtendedFrameMCW	8	0
MaxAssignmentSizeSISOTDD44	8	0
MaxAssignmentSizeSCWTDD44	8	0
MaxAssignmentSizeMCWTDD44	8	0
MaxAssignmentSizeSISOTDD63	8	0
MaxAssignmentSizeSCWTDD63	8	0
MaxAssignmentSizeMCWTDD63	8	0

MaxAssignmentSizeSixInterlaceSISO

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame SISO assignment in 6-interlace structure. Note that this also includes SCW and MCW assignments with rank 1.

MaxAssignmentSizeSixInterlaceSCW

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame SCW assignment in 6-interlace structure. Note that this includes SCW assignments with rank up to and including MaxSCWStreams.

MaxAssignmentSizeSixInterlaceMCW

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame MCW assignment in 6-interlace structure. Note that this includes MCW assignments with number of layers up to and including MaxMCWLayers.

MaxAssignmentSizeEightInterlaceSISO

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame SISO assignment in 8-interlace structure. Note that this also includes SCW and MCW assignments with rank 1.

MaxAssignmentSizeEightInterlaceSCW

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame SCW assignment in 8-interlace structure. Note that this includes SCW assignments with rank up to and including MaxSCWStreams.

MaxAssignmentSizeEightInterlaceMCW

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame MCW assignment in 8-interlace structure. Note that this includes MCW assignments with number of layers up to and including MaxMCWLayers.

MaxAssignmentSizeExtendedFrameSISO

This field shall be set to the maximum assignment size that the access terminal can decode for an extended frame SISO assignment. Note that this also includes SCW and MCW assignments with rank 1.

MaxAssignmentSizeExtendedFrameSCW

This field shall be set to the maximum assignment size that the access terminal can decode for an extended frame SCW assignment. Note that this includes SCW assignments with rank up to and including MaxSCWStreams.

MaxAssignmentSizeExtendedFrameMCW

This field shall be set to the maximum assignment size that the access terminal can decode for an extended frame SISO MCW assignment. Note that this includes MCW assignments with number of layers up to and including MaxMCWLayers.

MaxAssignmentSizeSISOTDD44

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame SISO assignment in TDD4:4 4-interlace structure. Note that this also includes SCW and MCW assignments with rank 1.

MaxAssignmentSizeSCWTDD44

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame SCW assignment in TDD4:4 4-interlace structure. Note that this includes SCW assignments with rank up to and including MaxSCWStreams.

MaxAssignmentSizeMCWTDD44

This field shall be set to the maximum assignment size that the access terminal can decode for a non-extended frame MCW assignment in TDD4:4 4-interlace structure. Note that this includes MCW assignments with number of layers up to and including MaxMCWLayers.

MaxAssignmentSizeSISOTDD63

This field shall be set to the maximum assignment size that the access terminal can decode in one FL PHY frame for a non-extended frame SISO assignment in TDD6:3 three-interlace structure. Note that this also includes SCW and MCW assignments with rank 1.

MaxAssignmentSizeSCWTDD63

This field shall be set to the maximum assignment size that the access terminal can decode in one FL PHY frame for a non-extended frame SCW assignment in TDD6:3 three-interlace structure. Note that this includes SCW assignments with rank up to and including MaxSCWStreams.

MaxAssignmentSizeMCWTDD63

This field shall be set to the maximum assignment size that the access terminal can decode in one FL PHY frame for a non-extended frame MCW assignment in TDD6:3 three-interlace structure. Note that this includes MCW assignments with number of layers up to and including MaxMCWLayers.

Commit	Soft allowed	Scope	Static

8.6.10 Protocol numeric constants

Protocol numeric constants are listed in Table 136.

Table 136—Protocol numeric constants

Constant	Meaning	Value
NFTCMPTType	Type field for this protocol.	See 16.1
NFTCMPBasic	Subtype field for this protocol.	0x00
MinHopportsPerNode	Minimum number of hop-ports in a channel tree node.	16
NFTCMultiHypothesis	Maximum number of MultipleHyothesisDecodeProfiles that may be supported by the access terminal.	16
NFTCMaxBCPacketSize	The largest size of the broadcast packet that the access terminal is required to support.	1024

The access network should not schedule a packet of size larger than NFTCMaxBCPacketSize with a broadcast MACID.

8.6.11 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

8.7 Basic Reverse Control Channel MAC Protocol

8.7.1 Overview

The Reverse Control Channel MAC defines the procedures for transmissions on the following Physical Layer channels:

- Reverse Pilot Channel (R-PICH)
- Reverse CDMA Dedicated Control Channel (R-CDCCH)
- Reverse OFDMA Dedicated Control Channel (R-ODCCH)
- Reverse Acknowledgement Channel (R-ACKCH)

The R CDCCH and R-ODCCH channels carry several logical channels, specifically:

- a) The R-CDCCH carries the following logical channels:
 - Reverse channel quality indicator channel (r-cqich)
 - Reverse request channel (r-reqch)
 - Reverse power amplifier headroom channel (r-pahch)
 - Reverse power spectral density channel (r-psdch)
- b) The R-ODCCH carries the following logical channels:
 - Reverse MIMO quality indicator channel (r-mqich)
 - Reverse channel quality indicator channel (r-cqich)
 - Reverse request channel (r reqch)
 - Reverse subband feedback channel (r-sfch)
 - Reverse beam feedback channel (r-bfch)

The R-PICH is a pilot channel. The R-CDCCH and R-ODCCH are CDMA and OFDMA dedicated control channels respectively. The r-cqich is used by the access terminal to transmit the FL quantized channel quality from different sectors to the access network. The r-reqch is used by the access terminal to request reverse link resources. The r-pahch is used by the access terminal to report its available transmit power. The r-psdch is used by the access terminal to report the path-loss differential between two sectors. The r-mqich carries MIMO channel quality information. The r-sfch is a feedback channel, that is used by the access terminal to transmit the quantized FL channel quality measured for a subband in the FL serving sector (FLSS). The r-bfch is a feedback channel, that is used by the access terminal to transmit the beam index, as well as supplemental channel quality information to enable SDMA transmission. SDMA transmission is defined in the Physical Layer Protocol. The R-ACKCH is used by the access terminal to acknowledge the Lower MAC packets transmitted on the forward link.

The access network maintains an instance of this protocol for every access terminal.

This protocol operates in one of two states:

- *Inactive State*: In this state, the access terminal is not assigned a MACID and shall not transmit on the Reverse Traffic Channel. When the protocol is in this state, it waits for an Activate command.
- *Active State*: In this state, the access terminal is assigned a MACID and may transmit data on the Reverse Traffic Channel.

The protocol states and the indications and events causing the transition between the states are shown in Figure 60.

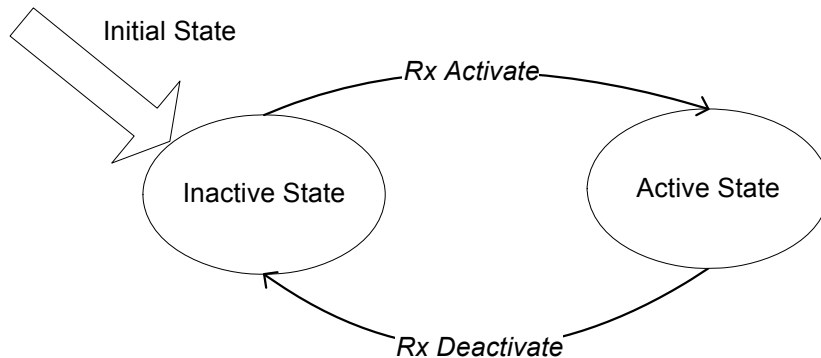


Figure 60—State Diagram for Tree Basic Reverse Control Channel MAC Protocol

This protocol uses the parameters and attributes shown in Table 137.

Table 137—Parameters and attributes for Reverse Control Channel MAC Protocol

Parameter Name	Where defined	Comments
ACKInterferenceOffset	OMP	ExtendedChannelInfo
ACKStepUpSize	Configuration attribute	—
ACKTargetCtoI	Configuration attribute	—
BFCHBeamCodeBookIndex	ASMP	Public data
BFCHCodebookType	ASMP	Public data
BFCHReportingMode	OMP	ExtendedChannelInfo
BRCHSubzoneCyclingEnabled	ASMP	Public data
CDMAInterlacesBitmap	OMP	ExtendedChannelInfo
CDCCHPeriod	ASMP	Public data
CDCCHPhase	ASMP	Public data
CDCCHSubsegmentIndex	ASMP	Public data
CQIPilotReportInterval	ASMP	Public data
CQIPilotReportPhase	ASMP	Public data
CQIReportInterval	ASMP	Public data
CQIReportPhase	ASMP	Public data
CQIReportRate	ASMP	Public data
CQITargetCtoI	Configuration attribute	—

**Table 137—Parameters and attributes for
Reverse Control Channel MAC Protocol (continued)**

Parameter Name	Where defined	Comments
CtrlAccessOffset	OMP	ExtendedChannelInfo
DefaultPQIValueForRLSS	Configuration attribute	—
HandoffCQICtoI	Configuration attribute	—
HandoffREQCtoI	Configuration attribute	—
InBandRequestTimerActive	RTC MAC	Public data
InterCellCQISwitchTimer	Configuration attribute	—
IntraCellCQISwitchTimer	Configuration attribute	—
IoTEnabled	OMP	ExtendedChannelInfo
MaxInterferenceOffset	Configuration attribute	—
MaxIOTCHErasures	Configuration attribute	—
MaxMCWLayers	FTC MAC	Configuration attribute
MaxNonFLSSCQIReportRate	ASMP	Public data
MaxNumLABs	OMP	ExtendedChannelInfo
MaxNumSharedLABs	OMP	ExtendedChannelInfo
MaxPAHCHReportRate	ASMP	Public data
MaxPQICHErasures	Configuration attribute	—
MaxPSDCHReportRate	ASMP	Public data
MaxRLPilotDifferenceForDFLSS	Configuration attribute	—
MaxRLPilotDifferenceForDRLSS	Configuration attribute	—
MaxRLPilotDifferenceForFLSS	Configuration attribute	—
MaxRLPilotDifferenceForRLSS	Configuration attribute	—
MaxSCWStreams	FTC MAC	Configuration attribute
MIMOMode	ASMP	Public data
MinBFCHReportRate	ASMP	Public data
MinChanDiffChange	ASMP	Public data

**Table 137—Parameters and attributes for
Reverse Control Channel MAC Protocol (continued)**

Parameter Name	Where defined	Comments
MinCQICHReportRate	ASMP	Public data
MinMQICHReportRate	ASMP	Public data
MinPAHChange	ASMP	Public data
MinPQIValue	Configuration attribute	—
MinRequestInterval	ASMP	Public data
MinSFCHReportRate	ASMP	Public data
NSSCQIPilotOffset	Configuration attribute	—
NullCQICtoI	Configuration attribute	—
NumAckableLABs	OMP	ExtendedChannelInfo
NumCDMASubsegments _j for $j = 1, \dots$, Bitcount(CDMAInterlacesBitmap)	OMP	ExtendedChannelInfo
NumDRCHSubzones	OMP	QuickChannelInfo
NumEffectiveAntennas	OMP	QuickChannelInfo
NumNULLCQIValidity	Configuration attribute	—
NumODCCHReportIndices	ASMP	Public data
NumODCCHReports	OMP	ExtendedChannelInfo
NumPrimaryRPICHSubsegments	ASMP	Public data
NumResourceSets	ASMP	Public data
NumResourceSubzones	ASMP	Public data
NumSecondaryRPICHSubsegments	ASMP	Public data
NumSFCHSubzones	ASMP	Public data
ODCCHPeriod	ASMP	Public data
ODCCHPhase	ASMP	Public data
ODCCHReportIndex _j , $j = 1, \dots$, NumODCCHReportIndices	ASMP	Public data
ODCCHReportRate	ASMP	Public data
ODCTargetCtoI	Configuration attribute	—
OpenLoopTransitionTime	Configuration attribute	—

**Table 137—Parameters and attributes for
Reverse Control Channel MAC Protocol (continued)**

Parameter Name	Where defined	Comments
PAHTargetCtoI	Configuration attribute	—
PDCABResourceSharingEnabled	OMP	ExtendedChannelInfo
PDCABSupported	Configuration attribute	FLCS MAC Protocol
PowerControlStepSize	Configuration attribute	—
PrimaryRPICHPeriod	ASMP	Public data
PrimaryRPICHPhase	ASMP	Public data
PrimaryRPICHSubsegment _j , $j = 1, \dots, \text{NumPrimaryRPICHSubsegments}$	ASMP	Public data
ProbePowerDensity	ACMP	Public data
PSDTargetCtoI	ASMP	Public data
RACKBandwidthFactor	OMP	ExtendedChannelInfo
REQQoSPowerBoost	Configuration attribute	—
REQTargetCtoI	Configuration attribute	—
ResourceChannelMuxMode	OMP	QuickChannelInfo
ResourceSetBitmap	ASMP	Public data
ResourceSetOffset	ASMP	Public data
ResourceSetSubzoneSpacing	ASMP	Public data
RoT _j , $j = 1, \dots, \text{Sum}(\text{NumCDMASubsegments})$	OMP	ExtendedChannelInfo
SecondaryRPICHSubsegment _j , $j = 1, \dots, \text{NumSecondaryRPICHSubsegments}$	ASMP	Public data
SingleServingSector	ASMP	Public data
SlowInterferenceOffset	OMP	ExtendedChannelInfo
StreamNNRequestPowerBoostAllowed	Configuration attribute	—
SubBandCQIOffset _j , $j = 1, \dots, \text{NumSubbands}$	ASMP	Public data
TotalNumSubcarriers	OMP	SystemInfo
VCQIMeasureInterval	ASMP	Public data

In Table 137, OMP, ACMP, and ASMP refer to the Overhead Messages Protocol, Access Channel MAC Protocol, and Active Set Management Protocol, respectively.

8.7.2 Primitives

8.7.2.1 Commands

This protocol defines the following commands:

- *ChangeFLSS*
- *ChangeRLSS*
- *Activate*
- *Deactivate*
- *ACKAssignment*

8.7.2.2 Return indications

The protocol returns the following indications:

- *FLSSChanged*
- *RLSSChanged*
- *DFLSSChanged*
- *DRLSSChanged*
- *SupervisionFailed*

8.7.2.3 Local common data

This protocol does not define any local common data.

8.7.2.4 Public data

This protocol shall make the following data public:

Subtype for this protocol.

- FLSS
- RLSS
- DFLSS
- DRLSS

8.7.3 Protocol data unit

This is a control protocol and does not carry any payload on the behalf of other protocols.

8.7.4 Procedures and messages for the InConfiguration instance of the protocol

8.7.4.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

8.7.4.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

8.7.4.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

8.7.5 Procedures for the InUse instance

8.7.5.1 Initialization and commit procedures

8.7.5.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the InActive State.
- The protocol at the access network shall enter the InActive State.

8.7.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3, when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

8.7.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

8.7.5.2 Command processing

8.7.5.2.1 Activate

If the protocol receives an *Activate* command in the Inactive State, the access terminal and the access network shall perform the following:

- Transition to the Active State

If the protocol receives this command in any other state, the command shall be ignored.

8.7.5.2.2 Deactivate

If the protocol receives a *Deactivate* command in the Active State,

- The access terminal shall cease transmitting the Reverse Control Channel and shall transition to the Inactive State.
- The access network shall cease monitoring the Reverse Control Channel from this access terminal and shall transition to the Inactive State.

If the protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

8.7.5.3 Inactive state

The access terminal shall not transmit on the physical channels governed by this protocol.

8.7.5.4 Active state

The access terminal shall transmit on the physical channels governed by this protocol according to the procedures in this subclause. This subclause makes frequent use of the following terms defined for each active access terminal:

active set	The set of sectors for which the access terminal has an assigned MACID.
RL interlace	A RL interlace i is defined as the set of all RL PHY frames with index k where $k \bmod N_{\text{RLInterlace}} = i$, where $N_{\text{RLInterlace}}$ equals 8 and 4 for the FDD and TDD mode, respectively.
DFLSS	(Desired Forward Link Serving Sector) The sector within the active set that the access terminal determines is the best sector for forward link data transmissions according to the procedures described in 8.7.5.4.3.
DRLSS	(Desired Reverse Link Serving Sector) The sector within the active set that the access terminal determines is the best sector for reverse link data transmissions according to the procedures described in 8.7.5.4.4.
FLSS	(Forward Link Serving Sector) The last DFLSS from which the access terminal successfully received a persistent FLAB with supplemental field '0', or a non-persistent FLAB with its own MACID (see 8.5.5.4.1.1.3.2).
RLSS	(Reverse Link Serving Sector) The last DRLSS from which the access terminal successfully received an RLAB with supplemental field '0' or a non-persistent RLAB with a unicast MACID (see 8.5.5.4.1.1.3.3).
CDMA Segment	Each RL PHY frame may contain a CDMA segment. RL PHY frame i contains a CDMA segment if the bit with index $i \bmod N_{\text{RLInterlace}}$ in $\text{CDMAInterlacesBitmap}$ is set to 1.
CDMA SubSegment Index	The index of a given SubSegment in CDMA Segments. These are indexed sequentially in time, with the first CDMA Subsegment in a group of $N_{\text{RLInterlace}}$ consecutive PHY frames $N_{\text{RLInterlace}} k, N_{\text{RLInterlace}} k + 1, \dots, N_{\text{RLInterlace}} k + N_{\text{RLInterlace}} - 1$ being indexed 0. A CDMA Subsegment in a later RL PHY frame in a group of $N_{\text{RLInterlace}}$ consecutive PHY frames has a larger CDMA SubSegment Index than a CDMA SubSegment in an earlier RL PHY frame in the same group of $N_{\text{RLInterlace}}$ PHY frames.
Reference Power Density	The notion of a reference power density is defined by the Physical Layer Protocol.

Also, the terms CoT and CtoI are used in this document to refer to the ratios of average carrier power to average thermal power and average carrier power to average thermal plus interference power, respectively. The exact methods for computing these average power quantities are out of the scope of this standard. Furthermore, the resources over which these average power quantities are computed may vary from one use to another, and are inferred from the context in which they are used.

8.7.5.4.1 Serving sector maintenance

When this protocol receives a *ChangeFLSS* command, it shall perform the following:

- Update the FLSS in the public data to be equal to the FLSS in the *ChangeFLSS* command
- Issue a *FLSSChanged* indication
- Update the DFLSS in the public data of this protocol to be equal to the FLSS in the *ChangeFLSS* command
- Issue a *DFLSSChanged* indication
- In addition, if the SingleServingSector bit for the FLSS in the *ChangeFLSS* command is set to '1', the protocol shall additionally perform the following:
 - Update the RLSS in the public data of this protocol, to be equal to the FLSS in the *ChangeFLSS* command
 - Issue a *RLSSChanged* indication
 - Update the DRLSS in the public data of this protocol, to be equal to the FLSS in the *ChangeFLSS* command
 - Issue a *DRLSSChanged* indication

When this protocol receives a *ChangeRLSS* command, it shall perform the following:

- Update the RLSS in the public data of this protocol
- Issue a *RLSSChanged* indication
- Update the DRLSS in the public data of this protocol
- Issue a *DRLSSChanged* indication
- If the SingleServingSector bit is set to '1' for the RLSS in the *ChangeRLSS* command, the protocol shall additionally perform the following:
 - Update the FLSS in the public data of this protocol, to be equal to the RLSS in the *ChangeRLSS* command
 - Issue a *FLSSChanged* indication
 - Update the DFLSS in the public data of this protocol, to be equal to the RLSS in the *ChangeRLSS* command
 - Issue a *DFLSSChanged* indication

8.7.5.4.2 Access terminal procedures for determining reverse link channel quality

For the purposes of determining the RL channel quality, the access terminal shall monitor F-PQICH from all sectors for which R-PICH is transmitted on the corresponding CDMA control subsegment.

For each sector, the access terminal shall interpret the most recent F-PQICH value intended for the access terminal in the FL shared control segment of that sector, as the post-processing received pilot carrier over thermal ratio at that sector. Here, the post-processing pilot carrier over thermal ratio is defined by Equation (11).

$$PQI = 10 \log_{10} \left(\left(\sum_{a=1}^{N_{RxAnt}} \sum_{i=1}^{N_{PICH}} P_{a,i} / N_{RxAnt} \right) / P_{Thermal} \right) \quad (11)$$

where

- N_{RxAnt} denotes the number of receiver antennas
- N_{PICH} denotes the number of modulation symbols used to transmit the pilot
- $P_{a,i}$ denotes the received pilot carrier power in mW on the a^{th} antenna and i^{th} modulation symbol
- $P_{Thermal}$ denotes the average thermal noise power per modulation symbol in mW

The mapping between the received F-PQICH value and the post-processing received pilot carrier over thermal ratio, labeled as $pCoT_{pp}$, is as described in Table 138. A received F-PQICH value of 0 indicates that the post-processing received pilot carrier over thermal ratio is less than -4 dB. Any other F-PQICH value indicates that the post-processing $pCoT$ is at least as much as indicated in the second column of Table 138. The access terminal shall define a pilot carrier over thermal ratio ($pCoT$) value based on the post-processing received pilot carrier over thermal ratio corresponding to the most recent F-PQICH value (PQI) as in Equation (12).

$$pCoT = pCoT_{pp} - 10\log_{10}(N_{PICH}) \quad (12)$$

where $N_{PICH} = 1024$ is the processing gain or the total number of modulation symbols used for the transmission of R-PICH.

Table 138—Mapping between F-PQICH value and $pCoT_{pp}$

Received F-PQICH Value	$pCoT_{pp}$
0	-6 dB
1	-4 dB
2	-2 dB
3	0 dB
4	2 dB
5	4 dB
6	5 dB
7	6 dB
8	7 dB
9	8 dB
10	9 dB
11	10 dB
12	11 dB
13	12 dB
14	13 dB
15	14 dB

The access terminal may choose to declare an erasure after decoding the F-PQICH value. The exact algorithms for decoding the F-PQICH and declaring erasures are beyond the scope of this standard. However, if the number of consecutive erasures of F-PQICH reports from a sector is smaller than $MaxPQICH_{Erasures}$, the access terminal shall use the most recent unerased F-PQICH report from that sector to compute the $pCoT$ value corresponding to that sector. If the access terminal has not received any F-PQICH report from its RLSS, or if the most recent $MaxPQICH_{Erasures}$ F-PQICH reports from RLSS are erased, then the access terminal shall set the corresponding $pCoT$ value as shown in Equation (13).

$$pCoT_{RLSS} = DefaultPQIValueForRLSS - 10\log_{10}(N_{PICH}) \quad (13)$$

where $DefaultPQIValueForRLSS$ is a configuration attribute of this protocol.

The access terminal shall set the pCoT value of an active set member s as shown in Equation (14).

$$pCoT_s = pCoT_{RLSS} - ChanDiff_s \quad (14)$$

where ChanDiffs is as defined in 8.8.5.5.1.9.1, if any of the following conditions is satisfied:

- The access terminal is not transmitting R-PICH to sector s .
- The access terminal has not received any F-PQICH report from sector s since it was added to the active set.
- The most recent MaxPQICHerasures F-PQICH reports from sector s are erased or carry PQI values that are smaller than MinPQIValue, where MaxPQIerasures and MinPQIValue are configuration attributes of this protocol.
- $ChanDiff_s > pCoT_{RLSS} - pCoT_s + MaxChanDiffImbalance$, where MaxChanDiffImbalance is a configuration attribute of this protocol.

The access terminal uses these pCoT values from different members of the active set for making FL and RL handoff request decisions. The access terminal also uses the pCoT values from its FL and RL serving sectors for determining the transmit power levels of different RL control channels, as well as RL traffic channel.

The access terminal is defined to be in RL lock with a sector if it receives at least one non-erased F-PQICH report over the MaxPQIerasures most recent F-PQICH opportunities from that sector, and the most recent non-erased F-PQICH report from that sector is greater than or equal to MinPQIValue.

8.7.5.4.3 Access terminal procedures for determining DFLSS

The access terminal monitors various air-link parameters to decide when to request handoff. Handoff is not completed until the AT receives assignment messages from the DFLSS and/or DRLSS.

The DFLSS chosen by the access terminal shall satisfy the following constraint:

$$pCoT_{Max} - pCoT_{DFLSS} < MaxRLPilotDifferenceForDFLSS$$

where $pCoT_{Max}$ is the largest pCoT value among active set members, and MaxRLPilotDifferenceForDFLSS is a configuration attribute of the RCC MAC protocol. The access terminal may choose any sector within its active set that satisfies the above constraint as its DFLSS. The access terminal may use received FL strengths to decide among the eligible sectors.

The access terminal shall set its DFLSS to be different from its FLSS if either of the following conditions is satisfied:

- $pCoT_{Max} - pCoT_{FLSS} > MaxRLPilotDifferenceForFLSS$, where MaxRLPilotDifferenceForFLSS is a configuration attribute of the RCC MAC protocol.
- The SingleServingSector bit for its FLSS is set to '1' and its DRLSS is different from its FLSS.

In either case, the access terminal may choose any sector within its active set other than its FLSS as its DFLSS subject to the constraint specified at the beginning of this subclause.

The access terminal should initiate a handoff request if its DFLSS is different from its FLSS. The handoff may be based on CQI repointing (i.e., sending a handoff r-cqich report on R-CDCCH of the DFLSS), or may be based on access (i.e., issue an *AccessChanneMAC.AttemptAccess* command with AccessType = 1). If the access terminal has more than one assigned ChannelBand on the DFLSS, the access terminal may initiate handoff request procedures on any of the ChannelBands.

The access terminal should use CQI repointing to indicate FL handoff request if it is in RL lock with the target sector, as defined in 8.7.5.4.2. If the access terminal is not in RL lock with the target sector, it should not use CQI repointing to indicate handoff request unless the R-CDCH transmissions to the target sector are reasonably time-aligned with the target sector. The access terminal shall use access based handoff if any of the following conditions are satisfied:

- AccessBasedHandoffRequired public data of the Connected State Protocol is equal to '1'. This behavior is useful for Semi-Connected Operation.
- AccessBasedHandoffRequired public data of the or Active Set Management Protocol is equal to '1' for the DFLSS. This behavior is useful for Multi-Carrier Operation.

The access terminal should change the DFLSS in accordance with the IntraCellCQISwitchTimer and InterCellCQISwitchTimer attributes. Information about which two sectors belong to the same cell is maintained by the Active Set Management Protocol.

The access terminal shall return a *SupervisionFailed* indication if the DFLSS is different from the FLSS for a period of time longer than FLHandoffSupervisionTimeout, which is a configuration attribute of this protocol.

8.7.5.4.4 Access terminal procedures for determining DRLSS

The access terminal monitors various air-link parameters to decide when to request handoff. Handoff is not completed until the AT receives assignment messages from the DFLSS and/or DRLSS.

The DRLSS chosen by the access terminal shall satisfy the following constraint:

$$pCoT_{Max} - pCoT_{DRLSS} < MaxRLPilotDifferenceForDRLSS$$

where $pCoT_{Max}$ is the largest $pCoT$ value among active set members, and $MaxRLPilotDifferenceForDRLSS$ is a configuration attribute of the RCC MAC protocol. The access terminal may choose any sector within its active set that satisfies the above constraint as its DRLSS. The access terminal may use RL pilot quality indications and interference offset reports received on F-PQICH and F-IOTCH from the active set members to decide among the eligible sectors.

The access terminal shall set its DRLSS to be different from its RLSS if any of the following conditions is satisfied:

- $pCoT_{Max} - pCoT_{RLSS} < MaxRLPilotDifferenceForRLSS$, where $MaxRLPilotDifferenceForRLSS$ is a configuration attribute of the RCC MAC protocol.
- The SingleServingSector bit for its RLSS is set to '1' and its DFLSS is different from its RLSS.
- The most recent MaxPQICHErasures F-PQICH reports from its RLSS are erased.

If any of the above conditions are satisfied, the access terminal may choose any sector within its active set other than its RLSS as its DRLSS subject to the constraint specified at the beginning of this subclause.

The access terminal should initiate a handoff request if its DRLSS is different from its RLSS. The requested handoff may be based on REQ repointing (i.e., sending a handoff r-reqch report on R-CDCH of the DRLSS), or may be based on access (i.e., issue an *AccessChannelMAC.AttemptAccess* command with *AccessType* = 1). If the access terminal has more than one assigned ChannelBand on the DRLSS, the access terminal may initiate handoff request procedures on any of the ChannelBands.

The access terminal should use REQ repointing to indicate RL handoff request if it is in RL lock with the target sector, as defined in 8.7.5.4.2. If the access terminal is not in RL lock with the target sector, it should not use REQ repointing to indicate handoff request unless the R-CDCH transmissions to the target sector

are reasonably time-aligned with the target sector. The access terminal shall use access based handoff if any of the following conditions are satisfied:

- AccessBasedHandoffRequired public data of the Connected State Protocol is equal to '1'. This behavior is useful for Semi-Connected Operation.
- AccessBasedHandoffRequired public data of the or Active Set Management Protocol is equal to '1' for the DRLSS. This behavior is useful for Multi-Carrier Operation.

The access terminal should change the DRLSS in accordance with the InterCellCQISwitchTimer and IntraCellCQISwitchTimer attributes. Information about which two sectors belong to the same cell is maintained by the Active Set Management Protocol.

The access terminal shall return a *SupervisionFailed* indication if the DRLSS is different from the RLSS for a period of time longer than RLHandoffSupervisionTimeout, which is a configuration attribute of this protocol.

8.7.5.4.5 Access terminal procedures for determining DSHOG

Access terminal's FLSS may belong to one or more softer handoff groups. The access terminal may autonomously select any softer handoff group that contains its FLSS as its Desired Softer Handoff Group (DSHOG) within its FLSS. The access terminal shall use the SHOG_ID of its DSHOG within its FLSS, in addition to the PilotID of its FLSS, as part of the scrambler seed to generate scrambling sequences for R-ODCCH and r-cqich transmissions on R-CDCCH to its FLSS.

Similarly, the access terminal's DFLSS may belong to one or more softer handoff groups. The access terminal may autonomously select any softer handoff group that contains its DFLSS as its DSHOG within its DFLSS. The access terminal shall use the SHOG_ID of its DSHOG within its DFLSS, in addition to the PilotID of its DFLSS, as part of the scrambler seed to generate scrambling sequences for R-ODCCH and r-cqich transmissions on R-CDCCH to its DFLSS.

8.7.5.4.6 Access terminal procedures for power control

The access terminal shall initialize the R-PICH transmission power per modulation symbol, P_{PICH} as

$$P_{PICH} = \text{ProbePower} + \text{CtrlAccessOffset}$$

where ProbePowerDensity parameter refers to the mean output power of the Access Channel probe at the end of the last successful Access Channel probe, and is public data of ACMP, and CtrlAccessOffset is public data of OMP.

8.7.5.4.6.1 Open-loop power adjustment

If OpenLoopTransitionTime is set to zero, the access terminal shall not perform the open-loop power adjustment procedures described in this subclause. Otherwise, the access terminal shall measure the mean received power during each superframe preamble of the RLSS, and compare it to the mean received power measured during the previous superframe preamble of the RLSS. The access terminal shall respond to the changes in this mean received power using a linear, time-invariant filter with the unit step response function shown in Equation (15).

$$s(t) = -\min\left\{\frac{t}{\text{OpenLoopTransitionTime}}, 1\right\} \quad (15)$$

where t denotes the time in PHY frames.

Since Equation (15) defines the step response for the transition, if multiple changes are measured within a single transition time, their effects on PPICH value shall be accumulated. Furthermore, any power adjustment based on the above rules shall be applied only at PHY frame boundaries, and shall be in addition to any changes dictated by the closed-loop power control algorithm described in 8.7.5.4.6.3.

8.7.5.4.6.2 Access terminal procedures for determining InterferenceOffset value

If IoTEEnabled bit is set to '1', the access terminal shall interpret the value received on F-IOTCH from RLSS or FLSS for subzone s as the interference offset value on that subzone of the corresponding sector. (The notion of a subzone is defined by the Physical Layer Protocol.) This value, denoted by $\text{InterferenceOffset}_{\text{FLSS},s}$ or $\text{InterferenceOffset}_{\text{RLSS},s}$, is expressed in units of dB, and is quantized using a quantization step size of 1 dB. The four-bit value, therefore, covers the range of 0 dB to 15 dB.

An F-IOTCH value received on FL PHY frame k shall be applicable starting RL PHY frame k_s . If DuplexingMode = FDD then $k_s = k + 3$, if DuplexingMode = TDD44, $k_s = k$ and if DuplexingMode = TDD63 $k_s = \lfloor k/2 \rfloor$. DuplexingMode is public data of the OMP.

The access terminal may choose to declare an erasure after decoding the F-IOTCH value. The exact algorithms for decoding the F-IOTCH and declaring erasures are beyond the scope of this standard. However, in the event of an erasure, the access terminal shall set the InterferenceOffsets value according to the following rules:

- If the number of consecutive erasures for a given F-IOTCH report is less than or equal to MaxIOTCHErasures , where MaxIOTCHErasures is a configuration attribute of this protocol, then the access terminal shall set the corresponding InterferenceOffset value to the most recent applicable unerased report.
- If the number of consecutive erasures for a given F-IOTCH report is greater than MaxIOTCHErasures , then the access terminal shall set the corresponding InterferenceOffset value to SlowInterferenceOffset value, which is public data of the OMP.

If IoTEEnabled bit is set to '0', the access terminal shall set all InterferenceOffset values (corresponding to different subzones) to SlowInterferenceOffset.

Furthermore, all the InterferenceOffset values are subject to a maximum value of $\text{MaxInterferenceOffset}$, which is a configuration attribute of this protocol. More precisely, if any InterferenceOffset value was larger than $\text{MaxInterferenceOffset}$, it shall be replaced by $\text{MaxInterferenceOffset}$.

8.7.5.4.6.3 Closed-loop power adjustment

The access terminal shall treat the power control bit intended for the access terminal that is sent over the F-PCCH of the RLSS as the power control command, and shall adjust the P_{PICH} value accordingly.

The P_{PICH} value is adjusted using the PowerControlStepSize parameters corresponding to the RLSS:

- When the PC bit transmitted over the F-PCCH of the RLSS is '1', the access terminal shall increase the P_{PICH} value by PowerControlStepSize dB.
- When the PC bit transmitted over the F-PCCH of the RLSS is '0', the access terminal shall decrease the P_{PICH} value by PowerControlStepSize dB.

These changes in P_{PICH} values shall be in addition to any changes dictated by the open-loop power control algorithm described in 8.7.5.4.6.1.

The output power per modulation symbol of reverse control channels for the RLSS and other members of the active set shall be derived from the P_{PICH} value, as described below.

The R-CDCCH transmit power per modulation symbol for an r-cqich report shall be computed according to the following rules:

- For an R-CDCCH transmission that carries an r-cqich report to FLSS or a non handoff r-cqich report (i.e., with the DFLSS flag set to '0') to RLSS in a disjoint link scenario (i.e., when RLSS is different from FLSS),

$$P_{CQICH,SS} = P_{PICH} - pCoT_{SS} + RoT_{SS} + CQITargetCtoI$$

where SS denotes the target serving sector (FLSS or RLSS), RoT_{SS} is the control RoT of the target serving sector on the CDMA subsegment over which the r-cqich report is being transmitted and is public data of the OMP, and CQITargetCtoI is a configuration attribute of this protocol.

- For an R-CDCCH transmission that carries a NULL r-cqich report to FLSS for flow control purposes,

$$P_{CQICH,NULL} = P_{PICH} - pCoT_{FLSS} + RoT_{FLSS} + NULLCQICtoI$$

where RoT_{FLSS} is the RoT of the FLSS on the CDMA subsegment over which the NULL r-cqich report is being transmitted and is public data of the OMP, and NullCQICtoI is a configuration attribute of this protocol.

- For an R-CDCCH transmission that carries a handoff r-cqich report to DFLSS (i.e., when the target sector for CQI is different from FLSS, and the DFLSS flag in the CQI report is set to '1'),

$$P_{CQICH,DFLSS} = P_{PICH} - pCoT_{DFLSS} + RoT_{DFLSS} + HandoffCQICtoI$$

where RoT_{DFLSS} is the RoT of DFLSS on the CDMA subsegment over which the r-cqich report is being transmitted and is public data of the OMP, and HandoffCQICtoI is a configuration attribute of this protocol.

- For an R-CDCCH transmission that carries an r-cqich report to any sector other than FLSS, RLSS, and DFLSS,

$$P_{CQICH,NSS} = P_{PICH} - pCoT_{NSS} + RoT_{NSS} + CQITargetCtoI$$

if $CQITargetCtoI - pCoT_{NSS} + RoT_{NSS} \leq NSSCQIPilotOffset$, where NSSCQIPilotOffset is a configuration attribute of this protocol. If this condition is not satisfied, i.e., if the total required boost with respect to R-PICH is larger than NSSCQIPilotOffset, then the access terminal shall not transmit the r-cqich report to this sector, i.e., it shall set $P_{CQICH,NSS} = 0$.

The R-CDCCH transmit power per modulation symbol for an r-reqch report shall be computed according to the following rules:

- For an R-CDCCH transmission that carries an r-reqch report to RLSS,

$$P_{REQCH,RLSS} = P_{PICH} - pCoT_{RLSS} + RoT_{RLSS} + REQTargetCtoI$$

- For an R-CDCCH transmission that carries a handoff r-reqch report to DRLSS (i.e., when the target sector for REQ is different from RLSS),

$$P_{REQCH,DRLSS} = P_{PICH} - pCoT_{DRLSS} + RoT_{DRLSS} + HandoffREQCtoI$$

Here RoT_{RLSS} and RoT_{DRLSS} are the RoT values of RLSS and DRLSS on the CDMA subsegment over which the r-reqch report is being transmitted and are public data of the OMP, and REQTargetCtoI and HandoffREQCtoI are configuration attributes of this protocol.

The R-CDCCH transmit power per modulation symbol for an r-pahch report to RLSS shall be computed according to the following rule:

$$P_{PAHCH, RLSS} = P_{PICH} - pCoT_{RLSS} + RoT_{RLSS} + PAHTargetCtoI$$

where RoT_{RLSS} is the control rise over thermal (RoT) of RLSS on the CDMA subsegment over which the r-pahch report is being transmitted and is public data of the OMP, and PAHTargetCtoI is a configuration attribute of this protocol.

The R-CDCCH transmit power per modulation symbol for an r-psdch report to RLSS shall be computed according to the following rule:

$$P_{PSDCH, RLSS} = P_{PICH} - pCoT_{RLSS} + RoT_{RLSS} + PSDTargetCtoI$$

where RoT_{RLSS} is the control rise over thermal (RoT) of RLSS on the CDMA subsegment over which the r-psdch report is being transmitted and is public data of the OMP, and PSDTargetCtoI is a configuration attribute of this protocol.

The R-ODCCH transmit power per modulation symbol shall be computed for the FLSS as

$$P_{ODCCH} = P_{PICH} - pCoT_{FLSS} + InterferenceOffset_{FLSS} + ODCTargetCtoI$$

where $InterferenceOffset_{FLSS}$ corresponds to the FLSS and is the largest InterferenceOffset value, as defined in 8.7.5.4.6.2, across all subzones, and ODCTargetCtoI is a configuration attribute of this protocol.

The R-ACKCH transmit power per modulation symbol shall be computed only for the FLSS according to the following rule:

$$P_{ACKCH} = P_{PICH} - pCoT_{FLSS} + SlowInterferenceOffset_{FLSS} + ACKInterferenceOffset_{FLSS} + ACKTargetCtoI + ACKCtoIAdjustment$$

where $SlowInterferenceOffset_{FLSS}$ and $ACKInterferenceOffset_{FLSS}$ correspond to the FLSS and is public data of the OMP, and ACKTargetCtoI is a configuration attribute of this protocol. The access terminal may use a non-zero ACKCtoIAdjustment value only if the access terminal is able to detect an ACK-to-NACK error event reliably with accuracy higher than 99%. ACKCtoIAdjustment shall be set to zero at the beginning of every Lower MAC packet transmission and shall increase by ACKStepUpSize dB every time an ACK-to-NACK error event is detected by the access terminal. The ACKStepUpSize parameter is a configuration attribute of this protocol.

If the access terminal is receiving multiple packets, then it shall maintain an independent value of ACKCtoIAdjustment for each packet. Then, the access terminal shall use the packet-specific value when transmitting R-ACKCH associated with that packet.

If the access terminal is unable to transmit the Physical Layer channels corresponding to the RCC MAC protocol and the RTC MAC protocol at the specified power due to access terminal's transmit power constraint, the access terminal shall transmit at the maximum transmit power. In this scenario, the access terminal may independently decide the relative transmission priority of these channels and gate-off the

transmission of one or more Physical Layer channels corresponding to RCC MAC protocol and RTC MAC protocol.

The access terminal shall not modify the power control variable PPICH after *AirLinkManagement.TunedAway* indication and until the *AirLinkManagement.TunedBack* indication. The access terminal shall not transmit the Physical Layer channels corresponding to the RCC MAC protocol and shall ignore the power control commands after *AirLinkManagement.TunedAway* indication and until the *AirLinkManagement.TunedBack* indication.

8.7.5.4.7 Procedures for determining CDMA subsegment coordinates

The CDMAInterlacesBitmap specifies which RL PHY frames within an interlace group of $N_{\text{RLInterlace}}$ consecutive PHY frames $N_{\text{RLInterlace}} k$, $N_{\text{RLInterlace}} k + 1$, ..., $N_{\text{RLInterlace}} k + 7$ contain CDMA segments. The NumCDMASubsegments gives the number of subsegments in each segment. These are sector-wide assignments. The indexing of the subsegments within a group of $N_{\text{RLInterlace}}$ consecutive PHY frames starts with 0 and proceeds sequentially. For instance, if the CDMAInterlacesBitmap is [0 1 1 0 0 1 0 1] and NumCDMASubsegments vector is [0 2 1 0 0 3 0 2], then the CDMA subsegments are indexed from 0 through 7. The subsegments in frames $N_{\text{RLInterlace}} k + 1$ (interlace 1) are indexed 0 and 1, the subsegment in frames $N_{\text{RLInterlace}} k + 2$ (interlace 2) is indexed 2, the subsegments in frames $N_{\text{RLInterlace}} k + 5$ (interlace 5) are indexed 3, 4, and 5 and the subsegments in frame $N_{\text{RLInterlace}} k + 7$ (interlace 7) are indexed 6 and 7. This is called the Overall Subsegment Index.

Correspondingly, for every Overall Subsegment Index, the Lower MAC layer has to compute the interlace index and the index of the subsegment within that interlace. For instance, subsegment 4 in the above example is referenced by interlaceindex 5 and subsegment index 1, that is, (5, 1). This is called the Subsegment Coordinate.

The procedure for computing Subsegment Coordinates for every Overall Subsegment Index is as follows:

- a) Add all the entries of the NumCDMASubsegments vector to get NumTotalSubsegments.
- b) Let CDMAOccurrence be the vector of indices where CDMAInterlacesBitmap is not 0.
- c) Initialize $k, l, m = 0$. Initialize $\text{FrameIndex} = \text{CDMAOccurrence}[0]$.
- d) Subsegment Coordinate for Overall Subsegment Index k is (FrameIndex, l)
 - 1) Increment k by 1.
 - 2) Increment l by 1. If $l = \text{NumCDMASubsegments}(\text{FrameIndex})$, $l = 0$. Increment m . $\text{FrameIndex} = \text{CDMAOccurrence}[m]$.
- e) Repeat step d) until $k = \text{NumTotalSubsegments}$.

8.7.5.4.8 Procedures for the R-PICH transmissions

The access terminal uses R-PICH to transmit pilots on CDMA subsegments. The access terminal shall transmit R-PICH to its RLSS and FLSS. The access terminal shall also transmit R-PICH to all other members of its active set that are mutually synchronous with its RLSS, as long as the total number of CDMA subsegments that the access terminal transmits on in a given interlace is less than or equal to the capability parameter MaxCDMASubsegmentsPerInterlace, which is a configuration attribute of this protocol.

The rules for transmitting R-PICH are described as follows:

- a) Each access terminal is assigned a NumPrimaryRPICHSubsegments and a list of PrimaryRPICHSubsegment $_j$, for $j = 1, \dots, \text{NumPrimaryRPICHSubsegments}$, a PrimaryRPICHPeriod and a PrimaryRPICHPhase by every member of the Active Set. The PrimaryRPICHSubsegment list is in terms of the Overall Subsegment Index described above. The PrimaryRPICHPeriod and PrimaryRPICHPhase are in units of $N_{\text{RLInterlace}}$ PHY frames. Let the RL

PH Frame index be $i = s \times N_{\text{RLPHYFrames}} + f$, where s is equal to superframe index mod 2^{33} , i.e., the 33 LSBs of the superframe index, $N_{\text{RLPHYFrames}}$ is the number RL PHY frames within a superframe, and f is the frame index within the current superframe.

$$\text{Define } j = \left\lfloor \frac{i}{N_{\text{RLInterlace}}} \right\rfloor.$$

Let $k = i \bmod N_{\text{RLInterlace}}$.

- b) If $j \bmod \text{PrimaryRPICHPeriod} = \text{PrimaryRPICHPhase}$ and $\text{CDMAInterlacesBitmap}(k) = 1$, then the primary R-PICH is transmitted on all subsegments s of PHY frame i for which (k, s) is the Subsegment Coordinate for some subsegment in the $\text{PrimaryRPICHSubsegment}_j$.
- c) An access terminal may also be assigned a $\text{NumSecondaryRPICHSubsegments}$ and a list of $\text{SecondaryRPICHSubsegment}_j$ for $j = 1, \dots, \text{NumSecondaryRPICHSubsegments}$. If the access terminal is assigned a list $\text{SecondaryRPICHSubsegment}_i$ from its FLSS or RLSS, then the access terminal shall also transmit secondary R-PICH according to the following rule: If the RL PHY frame index is i , let $k = i \bmod N_{\text{RLInterlace}}$. If $\text{CDMAInterlacesBitmap}(k) = 1$, then the secondary R-PICH is transmitted on all subsegments s of PHY frame i for which (k, s) is the Subsegment Coordinate for some subsegment in the list $\text{SecondaryRPICHSubsegment}_j$.

The access terminal shall transmit at most one instance of R-PICH on any CDMA subsegment of any PHY frame, i.e., if based on the above rules for primary and secondary R-PICH transmissions to different members of the active set, multiple transmissions happen to be on the same subsegment, only one instance of R-PICH shall be transmitted on that subsegment.

The RCC MAC Protocol shall provide the Physical Layer Protocol with P_{PICH} , and the target sector for each RPICH transmission. The P_{PICH} is determined according to the rules in 8.7.5.4.6.3.

The access terminal shall not transmit the R-PICH after *AirLinkManagement.TunedAway* indication and until the *AirLinkManagement.TunedBack* indication.

8.7.5.4.9 Procedures for the R-CDCCH physical channel

The R-CDCCH physical channel carries a 10-bit payload and transports the following logical channels: $r\text{-cqich}$, $r\text{-reqch}$, $r\text{-pahch}$, and $r\text{-psdch}$. The 10-bit payload may also be interpreted as an integer between 0 and 1023, with the leftmost bit being the MSB and the rightmost bit being the LSB. The payload for these logical channels shall be 5, 6, 6, and 4 bits respectively. Thus, these channels shall take 32, 64, 64, and 16 values respectively. The mapping of the value of each logical channel to the 10-bit R-CDCCH payload is according to the procedure described below. Payloads of distinct logical channels map to distinct R-CDCCH payloads.

- An $r\text{-cqich}$ value of i shall be mapped to an R-CDCCH value of $i + \text{MinCQIPayloadIndex}$.
- An $r\text{-reqch}$ value of i shall be mapped to an R-CDCCH value of $i + \text{MinREQPayloadIndex}$.
- An $r\text{-pahch}$ value of i shall be mapped to an R-CDCCH value of $i + \text{MinPAHPayloadIndex}$.
- An $r\text{-psdch}$ value of i shall be mapped to an R-CDCCH value of $i + \text{MinPSDPayloadIndex}$.

The values for $\text{MinCQIPayloadIndex}$, $\text{MinREQPayloadIndex}$, $\text{MinPAHPayloadIndex}$, and $\text{MinPSDPayloadIndex}$ are given in Table 139.

Table 139—Minimum Payload Index for different logical channels on the R-CDCCH

Logical channel	Minimum payload index	Physical channel payload range
r-cqich	MinCQIPayloadIndex = 0	0–31
r-reqch	MinREQPayloadIndex = 32	32–95
r-pahch	MinPAHPayloadIndex = 96	96–159
r-psdch	MinPSDPayloadIndex=160	160–175

In any frame, more than one R-CDCCH transmission may take place. For each R-CDCCH transmission, this protocol shall provide the Physical Layer Protocol with the PilotID of the target sector and the MACID of the access terminal within that sector.

The different R-CDCCH transmissions may correspond to different logical channels. The procedures for the logical channels are stated in later subclauses.

8.7.5.4.9.1 R-CDCCH reporting rules

If the access terminal is not transmitting R-PICH to any member of its active set, it shall not transmit any CDCCH report to that sector. The rules for reporting CDCCH are as follows:

- Each access terminal is assigned one CDMA subsegment for CDCCH reports, given by CDCCHSubsegmentIndex, as well as a CDCCHPeriod and a CDCCHPhase parameter, from every member of its active set. These parameters are public data of the ASMP. The CDCCHSubsegmentIndex specifies a CDMA subsegment in terms of its Overall Subsegment Index, as defined in 8.7.5.4.7, and CDCCHPeriod and CDCCHPhase are defined in terms of groups of $N_{\text{RLInterlace}}$ PHY frames.
- Let (k, s) denote the Subsegment Coordinate of CDCCHSubsegmentIndex, where the notion of Subsegment Coordinate is defined in 8.7.5.4.7. Let the RL PHY frame index be $p = s \times N_{\text{RLPHYFrames}} + f$, where s is equal to superframe index mod 2^{33} , i.e., the 33 LSBs of the superframe index, $N_{\text{RLPHYFrames}} = 25$ is the number of RL PHY frames within a superframe, and f is the frame index within the current superframe.

Define $q = \left\lfloor \frac{p}{N_{\text{RLInterlace}}} \right\rfloor$, and $r = p \bmod N_{\text{RLInterlace}}$.

If $r = k$ and $q \bmod \text{CDCCHPeriod} = \text{CDCCHPhase}$, then an R-CDCCH transmissions may be made on CDMA subsegment s on PHY frame p .

Specific rules for reporting each logical channel on R-CDCCH are described later in their respective subclauses.

8.7.5.4.10 Procedures for the R-ODCCH physical channel

This physical channel shall carry the following logical channels: r-mqich, r-cqich, r-reqch, r-sfch, and r-bfch. The rules for the R-ODCCH reports are as follows:

- The variable NumODCCHReports in the OMP defines the total number of segments allocated to OFDMA control over a set of $N_{\text{RLInterlaces}}$ consecutive PHY frames $N_{\text{RLInterlaces}}$ k , $N_{\text{RLInterlaces}}$ $k + 1$, ..., $N_{\text{RLInterlaces}}$ $k + N_{\text{RLInterlaces}} - 1$.

- b) Every PHY frame carrying an R-ODCCH report shall contain at least four R-ODCCH reports. A set of $N_{\text{RLInterlaces}}$ consecutive PHY frames shall contain at least 16 ODCCH reports. The Lower MAC assigns four reports to each interlace in round-robin fashion. The $N = \text{NumODCCHReports}$ are split across the interlaces as follows. The number of reports assigned to interlace j , N_j is given by $4 \times \lceil N / (4 \times N_{\text{RLInterlace}}) \rceil$ for $j < (N/4) \bmod N_{\text{RLInterlaces}}$, and $4 \times (\lceil N/4 \times N_{\text{RLInterlace}} \rceil - 1)$ otherwise.
- c) The reports within $N_{\text{RLInterlaces}}$ consecutive PHY frames $N_{\text{RLInterlaces}} k$, $N_{\text{RLInterlaces}} k + 1$, ..., $N_{\text{RLInterlaces}} k + N_{\text{RLInterlaces}} - 1$, are indexed by the Overall Report Index, which takes values from 0 to $\text{NumODCCHReports} - 1$. Each report will have a Frame Index that takes values from 0 to $N_{\text{RLInterlaces}} - 1$ and gives the index within the group of $N_{\text{RLInterlaces}}$ consecutive PHY frames of the PHY frame on which it occurs. Reports will also have a Subindex that specifies the report index within the PHY frame on which it occurs. Thus, every Overall Report Index can also be specified by a Report Coordinate, given by (Frame Index, Subindex). For example in the FDD mode, report with Overall Report Index 3 will have Report Coordinate (0, 3), Overall Report Index 22 will have Report Coordinate (5, 2), and Overall Report Coordinate 32 will have Report Coordinate (0, 4). More generally, consider Overall Report Index i .

$$\text{Let } j = \left\lfloor \frac{i}{4 \times N_{\text{RLInterlace}}} \right\rfloor, \text{ let } k = i \bmod N_{\text{RLInterlaces}}.$$

$$\text{Let } l = \frac{k N_{\text{RLInterlace}}}{4 \times N_{\text{RLInterlace}}}, \text{ let } m = k \bmod \frac{4 \times N_{\text{RLInterlace}}}{N_{\text{RLInterlace}}}.$$

Then, the Overall Report Index i refers to Report Coordinate $(l, \frac{4 \times N_{\text{RLInterlace}} \times j}{N_{\text{RLInterlace}}} + m)$.

8.7.5.4.10.1 R-ODCCH report

R-ODCCH shall have a payload of 22 bits and a header of 3 bits that specifies the mode of operation and the logical channels. The access terminal may choose the mode of operation autonomously at each instance subject to the constraints specified in each of the subclauses on logical channels. The following modes shall be supported (see Table 140).

Table 140—Payload on the R-ODCCH channel

Header value	r-cqich (4 bits)	r-rqch (6 bits)	r-sfch (8 bits)	r-bfch (8 bits)	scw r-mqich (7 bits)	mcw r-mqich (4 bits/layer)	Reserved bits
000	1	0	1	1	0	0	2
001	1	0	2	0	0	0	2
010	1	1	1	0	0	0	4
011	1	1	0	1	0	0	4
100	1	0	1	0	1	0	4
101	1	0	0	1	1	0	3
110	1	1	0	0	1	0	5
100	1	0	0	0	0	4	variable
101	1	1	0	0	0	3	variable
110	0	1	0	0	0	4	variable

Depending on the number of layers in the mcw r-mqich transmission, the number of reserved bits shall vary. Their number will be chosen so as to ensure that the total payload is 22 bits. The reserved bits shall be transmitted as '0'. The order of transmission of these bits is as indicated in Table 140 from left to right. For instance, if the mode is indicated by 001, then bits 0–3 shall correspond to the r-cqich report, bits 4–11 shall correspond to the first r-sfch report, bits 12–19 to the second r-sfch report, and bits 20–21 shall be the reserved bits, set to '00'. Bits for each logical channel shall be read from left to right as MSB to LSB.

8.7.5.4.10.1.1 R-ODCCH reporting rules

The rules for reporting ODCCH are as follows:

- a) Each user is assigned a list OFDMA Report indices, called ODCCHReportIndex, through the ASMP in which each entry is between 0 and NumOFDMAReports – 1. The length of this list is NumODCCHReportIndices. Each user is also assigned an ODCCHPeriod and ODCCHPhase through the ASMP. The Report Coordinate for each Report Index on the list is calculated as explained above. Consider an RL PHY frame with index p.

$$\text{Let } q = \left\lfloor \frac{p}{N_{\text{RLInterlace}}} \right\rfloor, \text{ let } r = p \bmod N_{\text{RLInterlace}}.$$

- b) If $q \bmod \text{ODCCHPeriod} = \text{ODCCHPhase}$, the R-ODCCH transmissions may be made to the FLSS on PHY frame p and Report Index y for every y such that (r, y) is the Report Coordinate for some Report Index in the ODCCHReportIndex list.
- c) The number of ODCCH reports transmitted by the access terminal shall be ODCCHReportRate over eight consecutive ODCCH report opportunities.

The RCC MAC Protocol shall provide the Physical Layer Protocol with the Report Index on that frame, P_{ODCCH} , PilotPN, SHOGID, and MACID for each R-ODCCH transmission. Here, the Report Index on a given frame refers to the second component of a Report Coordinate, as defined in 8.7.5.4.10, for which the first component is equal to the interlace index of that frame (i.e., frame index mod 8). The SHOGID shall be set to the index of the desired softer handoff group, and shall be set to zero if the access terminal does not wish to request FL transmission from a softer handoff group.

8.7.5.4.11 Procedures for the r-cqich logical channel

The access terminal measures the FL channel quality and transmits the quantized channel quality to the access network using the r-cqich. The computation and signaling method are a function of the ODCCHReportIndex list if the report occurs on the R-ODCCH or CDCCHSubsegmentIndex if the report occurs on the R-CDCCH. In addition, for reports carried on the R-CDCCH, the following parameters are required: CQIReportInterval, CQIReportPhase, CQIPilotReportInterval, CQIPilotReportPhase, CQIReportRate, and MaxNonFLSSCQIReportRate. The CQIReportInterval, CQIReportPhase, CQIPilotReportInterval, CQIPilotReportPhase are defined in terms of CDCCHPeriod. These are assigned to the access terminal by each member of the active set. The CQICHPIlot is used to transmit a known sequence and is transmitted less often than the CQICHCTRL, which contains the actual CQI information.

The RCC MAC Protocol shall provide the Physical Layer Protocol with the P_{CQICH} , PilotID, SHOGID, and MACID for each r-cqich transmission that occurs on R-CDCCH. The PilotID and MACID may correspond to the target sector for the CQI report, as discussed in 8.7.5.4.11.1. The P_{CQICH} is required for transmissions on R-CDCCH and is determined according to the rules in 8.7.5.4.6.3. The SHOGID shall be set to the index of the desired softer handoff group, and shall be set to zero if the access terminal does not wish to request FL transmission from a softer handoff group.

8.7.5.4.11.1 Reporting rules

A detailed description of the different CQI reports referred to in this subclause can be found in 8.7.5.4.11.2.

The r-cqich reports to sectors other than the FLSS shall be made on R-CDCCH. The r-cqich reports to the FLSS shall be made on R-ODCCH if the access terminal has an assigned ODCCHReportIndex list from the FLSS.

The following rules apply to r-cqich transmissions made on R-CDCCH:

- a) The access terminal shall transmit r-cqich only on CDMA subsegments specified by CQISubsegmentIndex CDCCHPeriod, and CDCCHPhase, as described in 8.7.5.4.9.1, assigned by the target sector. The MACID and PilotID of the target sector that are required for this shall be provided to the RL PHY protocol by the MAC protocol. Rules regarding the period of transmission and the phase are stated below. In these rules, all parameters correspond to those of the target sector.
 - 1) Let (a, b) denote the Subsegment Coordinate of CDCCHSubsegmentIndex, where the notion of Subsegment Coordinate is defined in 8.7.5.4.7. Let the RL PHY frame index be $p = s \times N_{RLPHYFrames} + f$, where s is equal to superframe index mod 233, i.e., the 33 LSBs of the superframe index, $N_{RLPHYFrames} = 25$ is the number of RL PHY frames within a superframe, and f is the frame index within the current superframe.
 Define $j = \left\lfloor \frac{p}{N_{RLInterlace}} \right\rfloor$, and $k = p \bmod N_{RLInterlace}$.
 - 2) If $j \bmod CQIPilotReportInterval \times CDCCHPeriod = CQIPilotReportPhase \times CDCCHPeriod + CDCCHPhase$ and $k = a$, then the access terminal shall transmit CQICHPILOT on subsegment b of PHY frame i.
 - 3) If $j \bmod CQIPilotReportInterval \times CDCCHPeriod \neq CQIPilotReportPhase \times CDCCHPeriod + CDCCHPhase$, $j \bmod CQIReportInterval \times CDCCHPeriod = CQIReportPhase \times CDCCHPeriod + CDCCHPhase$, and $k = a$, the access terminal may transmit CQICHCTRL on subsegment b of PHY frame i.
- b) The number of CQICHCTRL transmissions to FLSS over eight consecutive r-cqich opportunities shall be equal to CQIReportRate.
- c) The number of CQICHCTRL transmissions to sectors that are not FLSS or DFLSS shall not exceed MaxNonFLSSCQIReportRate in any consecutive set of eight superframes.

The rules for transmitting r-cqich on the R-ODCCH Physical Channel are as follows:

- The r-cqich report may only be sent on segments whose indices occur on the ODCCHReportIndex list.
- The access terminal shall send at least MinCQICHReportRate r-cqich reports over a set of eight consecutive ODCCH report opportunities.
- The access terminal shall send the CQIPilot reports on the R-CDCCH even when it has an R-ODCCH assignment.

The following rules apply to r-cqich transmissions made on R-CDCCH during handoff. Handoff CQI is defined as an r-cqich transmission to a non-FLSS sector with DFLSS = 1. The access terminal shall have a spacing of at least $N_{Backoff}$ r-cqich transmission opportunities between two successive handoff CQI transmissions. The rules for calculating $N_{Backoff}$ are given as follows:

- The access terminal shall initialize $M = 1$ when either of the following conditions is true:
 - There is a change in DFLSS
 - DFLSS and FLSS are different for a time period larger than CQIBackoffResetTime
- The access terminal shall calculate $N_{Backoff}$ as follows:
 - Set $p = CQIBackoffFactor^M$
 - Generate a random variable that takes value $n \geq 0$ with probability $p(1 - p)^n$
 - Calculate n_{max} as the smallest value of n for which $p(1 - p)^n$ is less than 0.01

- Set N_{Backoff} to $\min(n, n_{\text{max}})$
- After sending each handoff CQI, the access terminal shall increment M

8.7.5.4.11.2 CQI report

The CQI reports are shown in Table 141.

Table 141—CQI reports

CQI Report	CQICH Pilot
	CQICH CTRL

The format for the CQICH Pilot report is shown in Table 142. The CQICH Pilot shall be sent on the R-CDCCH even when it has an R-ODCCH assignment.

Table 142—Format for CQICH Pilot report

Field	Length (bits)
ReservedValue	5

ReservedValue The ReservedValue is set to 0.

The format for the CQICH CTRL is shown in Table 143.

Table 143—Format for CQICH CTRL report

Field	Length (bits)
CQIValueSISO	4
DFLSSFlag	1

CQIValueSISO Indicates FL SISO CQI value. See 8.7.5.4.11.3 for details.

DFLSSFlag If the target sector is the current DFLSS, the DFLSSFlag bit shall be set to 1; otherwise, the DFLSSFlag bit shall be set to 0. This field is absent when r-cqich is carried on the R ODCCH.

8.7.5.4.11.3 CQIValueSISO

Based on an unrestricted observation interval, the access terminal shall report the highest tabulated non-precoded CQI value, CQIValueSISO such that a packet transmitted by the access network to the access terminal over all of the hop-ports sent at Reference Power Density using the specified packet format and the specified number of FL-PHY frames, terminating 1 PHY frame before the start of the PHY frame in which the r-cqich is transmitted, would result in a packet error probability ≤ 0.01 . The CQI mappings are shown in Table 144. The notion of non-precoded CQI value is defined in 8.7.5.4.12.

The r-cqich uses only 4 bits for reporting the CQI. The 5-bit values indicated in the table are used by r-mqich logical channel as described in 8.7.5.4.13.2.

A 4-bit CQI value of '0000' is called the NULL CQI indicates one of the following two conditions:

- It indicates to the access network that it should stop sending data to the access terminal on the FL. This is referred to as flow control.
- None of the packet formats indicated by a non-NULL CQI value can be supported at target FER.

NULL CQI reports follow the following rules:

- a) If the NULL CQI value is being used for flow control, then it only be transmitted on odd CQI reports, as defined below.
- b) If NULL CQI value is being used for flow control, then even reports, as defined below, shall be used to report regular CQI.
- c) If the access network receives a NULL CQI on an odd CQI report, it shall start a timer with duration NumNULLCQIValidity PHY frames, where NumNULLCQIValidity is a configuration attribute of this protocol. The access network shall not schedule data to the access terminal while this timer is active. The access network shall disable the timer if it receives a non-NULL CQI value on an odd CQI report opportunity.

Here, odd and even CQI reports are defined as follows. Let the RL PHY frame index be $p = s \times N_{\text{RLPHYFrames}} + f$, where s is equal to superframe index mod 233, i.e., the 33 LSBs of the superframe index, $N_{\text{RLPHYFrames}}$ is the number of RL PHY frames within a superframe, and f is the frame index within the current superframe. A CQI report transmitted on frame p is said to be an even CQI report if

$$\left\lfloor \frac{p}{N_{\text{RLInterlace}} \times \text{CDCCHPeriod} \times \text{CQIReportInterval}} \right\rfloor \bmod 2 = 0,$$

and it is said to be an odd CQI report, otherwise.

Table 144—CQI Mapping to the forward link packet format and number of forward link PHY frames

4-bit CQI value	5-bit CQI value	FL packet format	Number of FL-PHY frames
0	0	Used for Flow Control or if the packet format indicated by no other CQI value can be supported with target PER	N/A
	1	0	6
	2	0	4
1	3	0	3
	4	1	5
2	5	1	4
	6	2	4
3	7	3	6
	8	3	5
4	9	3	4
	10	5	6

Table 144—CQI Mapping to the forward link packet format and number of forward link PHY frames (continued)

4-bit CQI value	5-bit CQI value	FL packet format	Number of FL-PHY frames
5	11	5	5
	12	7	5
6	13	8	5
	14	4	2
7	15	7	3
	16	10	4
8	17	6	2
	18	13	5
9	19	14	5
	20	13	4
10	21	9	2
	22	10	2
11	23	13	3
	24	14	3
12	25	12	2
13	26	13	2
14	27	9	1
15	28	14	2
—	29	N/A	N/A
—	30	N/A	N/A
—	31	N/A	N/A

8.7.5.4.11.4 Vector channel quality indicator (VCQI) support

Vector channel quality indicator (VCQI) is designed to support fractional frequency reuse, where interference coordination is implemented among multiple sectors. The transmission rules for the VCQI message are defined by the ASMP. Resource sets are defined in each sector for transmissions to and from access terminals in different radio conditions. A resource set is a union of (subzone, interlace) pairs, where subzone is a set of hop-ports defined in the Physical Layer Protocol. A forward link resource set has the same number of subzones of each type (DRCH/BRCH) on every interlace where this resource set is present. The ResourceSets are defined by the ASMP, using the variables NumResourceSets, ResourceSetBitmap, BRCHSubzoneCyclingEnabled, ResourceSetSubzoneSpacing, NumResourceSubzones, and ResourceSetOffset, which are part of its public data.

Define CQIValueSCWFrame[ResourceSetIndex][Rank] to indicate the FL SCW CQI value for the given resource set index and Rank; CQIValueMCWFrame[ResourceSetIdx][Layer] to indicate the FL MCW CQI

value for each MIMO layer. Note that the resource set is composed of a combination of interlaces and subzones as defined in 8.5.5.4.1.1.3.2.1.

Based on an unrestricted observation interval, the access terminal shall compute the $CQIValueSCWFrame[ResourceSetIndex][Rank]$ or the $CQIValueMCWFrame[ResourceSetIndex][Layer]$ depending on the $MIMOMode$, which is public data of the ASMP, such that a packet transmitted by the access network to the access terminal over all of the hop-ports in the resource set, sent at Reference Power Density using the specified packet format and the specified number of FL-PHY frames, terminating 1 PHY frame before the start of the PHY frame where $CQIValueSCWFrame[ResourceSetIndex][Rank]$ or the $CQIValueMCWFrame[ResourceSetIndex][Layer]$ is computed, would result in a packet error probability ≤ 0.01 . The CQI mappings are shown in Table 144.

If $MIMOMode$ is SCW, the access terminal shall compute the report $VCQIValueSCW[ResourceSetIndex][Rank]$ for each Rank, by averaging the $CQIValueSCWFrame$ computed for that particular resource set and Rank, over an averaging interval specified by $VCQIMeasureInterval$, terminating 1 PHY frame before the VCQI message is reported. Note that for a non-MIMO capable access terminal, the Rank index is 0; for MIMO capable access terminal, the Rank index ranges from 0 to $MaxSCWStreams - 1$.

If $MIMOMode$ is MCW, the access terminal shall compute the report $VCQIValueMCW[ResourceSetIndex][Layer]$ by averaging the $CQIValueMCWFrame$ computed for that particular resource set and MIMO layer, over an averaging interval specified by $VCQIMeasureInterval$, terminating 1 PHY frame before the VCQI message is reported. Note that for a non-MIMO capable access terminal, the Layer index is 0; for MIMO capable access terminal, the Layer index ranges from 0 to $MaxMCWLayers - 1$.

8.7.5.4.11.5 CellNullCQISupport

CellNullCQI shall be computed according to non-precoded FL SISO CQI with interference measured over F-CNCH.

Based on an unrestricted observation interval, the access terminal shall compute the highest tabulated non-precoded CellNullCQIValue, according the same rule as $CQIValueSISO$, except that the interference is measured over F-CNCH.

The CellNullCQI is computed as the average CellNullCQIValue over an averaging interval specified by $CellNullCQIReportInterval$, terminating 1 PHY frame before the CellNullCQI is reported.

8.7.5.4.12 Precoded and non-precoded CQI value

In this subclause, precoded CQI refers to a CQI value that the access terminal computes assuming that the access network uses the upper-left $M \times R$ submatrix of the precoding matrix indicated by the BeamIndex field in the r-bfch report, for data transmissions to that access terminal. Each column of the $M \times R$ submatrix is interpreted as a linear combination of the first M effective antennas. Here $M = \min(NumEffectiveAntennas, MaxPrecodedTxAntennas)$, where $NumAntennas$ is the number of rows of the precoding matrix, $NumEffectiveAntennas$ is the number of effective antennas defined by the OMP, $MaxPrecodedTxAntennas$ is a configuration attribute of the FTC MAC protocol, and R is the rank of the CQI feedback. R is equal to one plus the value of the Rank field for SCW MIMO CQI, and R is equal to the number of layers for MCW MIMO CQI.

The interpretation of the non-precoded CQI depends on the resource channel multiplexing mode.

- If $ResourceChannelMuxMode = 1$ or $NumDRCHSubzones > 0$, non-precoded CQI refers to a CQI value that the access terminal computes assuming that the access network uses the first R effective

antennas for data transmissions to that access terminal. Here, $R = 1$ for SISO CQI value. For MIMO CQI, R is as defined above in the case of precoded CQI.

- If ResourceChannelMuxMode = 2 and NumDRCHSubzones = 0, non-precoded CQI refers to an average CQI value that should be computed assuming that different matrices randomly selected with uniform probability from the Random Unitary Ensemble with parameters (M, R) , as defined by the Physical Layer protocol, are applied to different time-frequency resources. These different unitary matrices are assumed to be statistically independent. The selection of the matrices is beyond the scope of this standard. Each column of the $M \times R$ matrix is interpreted as a linear combination of the first M effective antennas. Here $M = \min(\text{NumEffectiveAntennas}, \text{MaxPrecodedTxAntennas})$, $R = 1$ for SISO CQI value, and for MIMO CQI, R is as defined above in the case of precoded CQI. Note that Random Unitary Ensemble is defined for CQI calculation only.

8.7.5.4.13 Procedures for the r-mqich logical channel

During MIMO operation, the access terminal measures the broadband FL channel quality and transmits the quantized channel quality to the FLSS using the r-mqich. When r-sfch is enabled or when both r-sfch and r-bfch are disabled, r-mqich shall carry the broadband non-precoded MIMO or STTD CQI value. Otherwise, the r-mqich shall carry the broadband precoded MIMO CQI value corresponding to the preferred beam index as specified in the r-bfch. The r-mqich logical channel shall be carried on the R-ODCCH Physical Channel. The r-mqich report depends on the value of the MIMOMode variable, which takes values MCW and SCW.

8.7.5.4.13.1 Reporting rules

The r-mqich report may only be sent on segments whose indices occur on the ODCCHReportIndex list. The access network ignores r-mqich reports if any of the following are satisfied:

- r-bfch is enabled, r-sfch is disabled, MIMOMode = 'SCW', and MaxSCWStreams = 1.
- r-bfch is enabled, r-sfch is disabled, MIMOMode = 'MCW', and MaxMCWLayers ≤ 1 .
- NumEffectiveAntennas = 1.
- MinMQICHReportRate = 0.

If none of the above conditions is true, then the access terminal shall send at least MinMQICHReportRate r-mqich reports over a set of eight consecutive ODCCH report opportunities.

8.7.5.4.13.2 MCQI report

The MCQIReport shall have two reporting modes, for SCW and MCW transmissions. When the MIMOMode is SCW or MCW, the r-mqich reporting mode shall be MQICHSCW or MQICHMCW, respectively.

The format for the MQICHSCW is in Table 145.

Table 145—Format for MQICHSCW report

Field	Length (bits)
MCQIValueSCW	5
Rank	2

MCQIValueSCW

When r-sfch is enabled or when both r-sfch and r-bfch are disabled, MCQIValueSCW indicates the broadband non-precoded FL MIMO SCW CQI value or FL STTD CQI value for the reported Rank as follows:

- If the number of effective transmit antennas is two:
 - If the value in the Rank field is 0:
 - If $\text{MaxSTTDRank} \geq 1$, MCQIValueSCW indicates the FL STTD CQI value;
 - otherwise, MCQIValueSCW indicates the FL MIMO SCW CQI value.
 - If the value in the Rank field is 1, MCQIValueSCW indicates the FL MIMO SCW CQI value
- If the number of effective transmit antennas is four:
 - If the value in the Rank field is 0:
 - If $\text{MaxSTTDRank} \geq 1$, MCQIValueSCW indicates the FL STTD Mode A CQI value;
 - otherwise, MCQIValueSCW indicates the FL MIMO SCW CQI value.
 - If the value in the Rank field is 1:
 - If $\text{MaxSTTDRank} = 2$, MCQIValueSCW indicates the FL STTD Mode B CQI value;
 - otherwise, MCQIValueSCW indicates the FL MIMO SCW CQI value.
 - If the value in the Rank field is 2, MCQIValueSCW indicates the FL MIMO SCW CQI value
 - If the value in the Rank field is 3, MCQIValueSCW indicates the FL MIMO SCW CQI value
- If the number of effective transmit antennas is not equal to 2 or 4:
 - MCQIValueSCW indicates the FL MIMO SCW CQI value for the reported rank
- Otherwise, when r-bfch is enabled and r-sfch is disabled, MCQIValueSCW indicates the broadband precoded FL MIMO SCW CQI value corresponding to the BeamIndex as specified in the r-bfch. If binary unitary precoding is used with STTD with antenna selection indication, then MCQIValueSCW indicates the CQI value computed assuming STTD over selected two antennas.

See 8.7.5.4.13.3 for details.

Rank Indicates one less than the desired number of MIMO layers in the FL MIMO SCW transmission.

The format for the MQICHMCW report is shown in Table 146. If $2 \leq \text{NumEffectiveAntennas} \leq 8$, the MQICHMCW report consists of four bits for each of the layers. Therefore, it consists of 4 bits or 8 bits or 12 bits or 16 bits if the number of layers is 1 or 2 or 3 or 4.

Table 146—Format for MQICHMCW report

Field	Length (bits)
MCQIValueMCWLayer1	4
MCQIValueMCWLayer2	4
MCQIValueMCWLayer3	4

Table 146—Format for MQICHMCW report (continued)

Field	Length (bits)
MCQIValueMCWLayer4	4

NOTE—Only as many fields as the number of layers have to be present.

MCQIValueMCWLayer1 When r-sfch is enabled or when both r-sfch and r-bfch are disabled, MCQIValueMCWLayer1 indicates the broadband non-precoded FL MIMO MCW layer 1 CQI value or FL STTD CQI value as follows:

- If the number of effective transmit antennas is two:
 - If the overall reported rank is one,
 - If $\text{MaxSTTDRank} \geq 1$, MCQIValueMCWLayer1 indicates the FL STTD CQI value.
 - Otherwise, MCQIValueMCWLayer1 indicates the FL MIMO MCW layer 1 CQI value.
 - If the overall reported rank is two, MCQIValueMCWLayer1 indicates the FL MIMO MCW layer 1 CQI value.
- If the number of effective transmit antennas is four:
 - If the overall reported rank is one,
 - If $\text{MaxSTTDRank} = 2$, MCQIValueMCWLayer1 indicates the FL STTD Mode A CQI value.
 - Otherwise, MCQIValueMCWLayer1 indicates the FL MIMO MCW layer 1 CQI value.
 - If the overall reported rank is two,
 - If $\text{MaxSTTDRank} = 2$, MCQIValueMCWLayer1 indicates the FL STTD Mode B CQI value.
 - Otherwise, MCQIValueMCWLayer1 indicates the FL MIMO MCW layer 1 CQI value.
 - If the overall reported rank is three or four, MCQIValueMCWLayer1 indicates FL MIMO MCW layer 1 CQI value.
- If the number of effective transmit antennas is not equal to two or four:
 - MCQIValueMCWLayer1 indicates FL MIMO MCW layer 1 CQI value.
- Otherwise, when r-bfch is enabled and r-sfch is disabled, MCQIValueMCWLayer1 indicates the broadband precoded FL MIMO MCW layer 1 CQI value corresponding to the BeamIndex as specified in the r-bfch.

See 8.7.5.4.13.3 for details.

MCQIValueMCWLayer2

When r-sfch is enabled or when both r-sfch and r-bfch are disabled, MCQIValueMCWLayer2 indicates the broadband non-precoded FL MIMO MCW layer 2 CQI value or FL STTD CQI value as follows:

- If the number of effective transmit antennas is two:
 - If the overall reported rank is two, MCQIValueMCWLayer2 indicates the FL MIMO MCW layer 2 CQI value.

- If the number of effective transmit antennas is four:
 - If the overall reported rank is two,
 - If $\text{MaxSTTD Rank} = 2$, $\text{MCQIValueMCWLayer2}$ indicates the FL STTD Mode B CQI value.
 - Otherwise, $\text{MCQIValueMCWLayer2}$ indicates FL MIMO MCW layer 2 CQI value.
 - If the overall reported rank is three or four, $\text{MCQIValueMCWLayer2}$ indicates FL MIMO MCW layer 2 CQI value.
- If the number of effective transmit antennas is not equal to two or four:
 - The $\text{MCQIValueMCWLayer2}$ indicates FL MIMO MCW layer 2 CQI value.
- Otherwise, when r-bfch is enabled and r-sfch is disabled, $\text{MCQIValueMCWLayer2}$ indicates the broadband precoded FL MIMO MCW layer 2 CQI value corresponding to the BeamIndex as specified in the r-bfch.

See 8.7.5.4.13.3 for details.

MCQIValueMCWLayer3 When r-sfch is enabled, $\text{MCQIValueMCWLayer3}$ indicates the broadband non-precoded FL MIMO MCW layer 3 CQI value.

Otherwise, when r-bfch is enabled and r-sfch is disabled, $\text{MCQIValueMCWLayer3}$ indicates the broadband precoded FL MIMO MCW layer 3 CQI value corresponding to the BeamIndex as specified in the r-bfch.

See 8.7.5.4.13.3 for details.

MCQIValueMCWLayer4 When r-sfch is enabled, $\text{MCQIValueMCWLayer4}$ indicates the broadband non-precoded FL MIMO MCW layer 4 CQI value.

Otherwise, when r-bfch is enabled and r-sfch is disabled, $\text{MCQIValueMCWLayer4}$ indicates the broadband precoded FL MIMO MCW layer 4 CQI value corresponding to the BeamIndex as specified in the r-bfch.

See 8.7.5.4.13.3 for details.

8.7.5.4.13.3 MCQIValue

Based on an unrestricted observation interval, the access terminal shall report the highest tabulated CQI value such that a packet transmitted by the access network to the access terminal over all of the hop-ports sent at Reference Power Density using the specified packet format and the specified number of FL-PHY frames, terminating one FL PHY frame before the start of the RL PHY frame in which the r-cqich is transmitted, would result in a packet error probability ≤ 0.01 . The CQI mappings are shown in Table 144. From this table, the column corresponding to the 5-bit values is used for the MCQISCW reports (MCQValueSCW) and the column corresponding to the 4-bit values is used for the MCQIMCW reports ($\text{MCQIValueMCWLayer}k$).

8.7.5.4.14 CQIMode

The access terminal shall determine that CQIMode is SISO if both of the following conditions are satisfied:

- MIMOMode is SCW and MaxSCWStreams in the FTC-MAC is set to 1.

- r-sfch is enabled or r-bfch is disabled.

The access terminal shall determine that CQIMode is SCW if MIMOMode is SCW and either of the following conditions is satisfied:

- MaxSCWStreams > 1.
- MaxSCWStreams = 1, r-sfch is disabled and r-bfch is enabled.

The access terminal shall determine that CQIMode is MCW if MIMOMode is MCW.

8.7.5.4.15 Procedures for the r-sfch logical channel

The access terminal uses the r-sfch to transmit the index and the quantized FL channel quality corresponding to the a group of subcarriers. Each such group is referred to as a subband, and has a corresponding SubbandIndex. The number of subbands is given by NumSubbands, where

$$\text{NumSubbands} = \left\lceil \frac{N_{\text{FFT}} - N_{\text{GUARD}}}{\text{FLSubzoneSize} \times \text{NumSFCHSubzones}} \right\rceil$$

Here, N_{FFT} is equal to TotalNumSubcarriers and N_{GUARD} is equal to NumGuardSubcarriers. FLSubzoneSize, TotalNumSubcarriers and NumGuardSubcarriers are parameters of the Overhead Messages Protocol and NumSFCHSubzones is a parameter of the ASMP.

The subband with SubbandIndex s has $N_{\text{SUBBAND}}(s)$ subcarriers, where

$$N_{\text{SUBBAND}}(s) = N_{\text{BLOCK}} \left\lceil \frac{N_{\text{FFT}} - N_{\text{GUARD}}}{N_{\text{BLOCK}} \times \text{NumSubbands}} \right\rceil,$$

if $s < ((N_{\text{FFT}} - N_{\text{GUARD}})/N_{\text{BLOCK}}) \bmod \text{NumSubbands}$ and

$$N_{\text{SUBBAND}}(s) = N_{\text{BLOCK}} \left\lfloor \frac{N_{\text{FFT}} - N_{\text{GUARD}}}{N_{\text{BLOCK}} \times \text{NumSubbands}} \right\rfloor$$

Here, N_{BLOCK} is a constant of the Physical Layer Protocol. The subcarriers are allocated to the subbands in sequential order i.e., the subband with SubbandIndex 0 comprises of subcarriers $N_{\text{GUARD}}/2$ through $N_{\text{GUARD}}/2 + N_{\text{SUBBAND}}(0) - 1$, the subband with SubbandIndex 1 comprises of subcarriers $N_{\text{GUARD}}/2 + N_{\text{SUBBAND}}(0)$ through $N_{\text{GUARD}}/2 + N_{\text{SUBBAND}}(0) + N_{\text{SUBBAND}}(1) - 1$, the subband with SubbandIndex 2 comprises of subcarriers $N_{\text{GUARD}}/2 + N_{\text{SUBBAND}}(0) + N_{\text{SUBBAND}}(1)$ through $N_{\text{GUARD}}/2 + N_{\text{SUBBAND}}(0) + N_{\text{SUBBAND}}(1) + N_{\text{SUBBAND}}(2) - 1$, etc.

The channel quality is transmitted differentially with respect to the broadband CQI. Thus, the r-sfch transmission, along with the r-cqich and r-mqich transmissions shall give the channel quality as described below.

8.7.5.4.15.1 Reporting rules

The reporting rules for r-sfch are as follows:

- a) The r-sfch report may only be sent on segments whose indices occur on the ODCCHReportIndex list.
- b) The access terminal shall send at least MinSFCHReportRate r-sfch reports over a set of eight consecutive ODCCH report opportunities. The access network shall ignore the r-sfch reports from

the access terminal if MinSFCHReportRate is equal to the value 0 (this case will be referred to as r-sfch being disabled).

The access terminal shall not transmit the r-sfch after *AirLinkManagement.TunedAway* indication and until the *AirLinkManagement.TunedBack* indication.

8.7.5.4.15.2 SFCH report

The r-sfch reports associated with r-cqich and r-mqich are shown in Table 147.

Table 147—SFCH Report Types

	CQIMode		
	SISO	SCW	MCW
SFCH Report	SFCHSISO	SFCHSCW	SFCHMCW

The format for the SFCH report is shown in Table 148.

Table 148—Format for SFCH report

Field	Length (bits)
SubBandIndex	4
SubBandCQIValueDelta	4

SubBandIndex Indicates the set of NumSFCHSubzones subzones for which the SubBandCQIValueSISO is reported. The access terminal shall report the value of SubBandIndex for one (in R-ODCCH reports with the header values of '000', '010', and '100') or two (in R-ODCCH reports with the header value of '001') subbands that have the highest values of SubbandCQIValue + SubBandCQIOffset_{SubBandIndex}, where SubBandCQIValue is as defined below, and SubBandCQIOffset_{SubBandIndex} is public data of the ASMP.

SubBandCQIValueDelta This field indicates the differential CQI Value for the reported SubBandIndex. This value gives the SubBandCQIValueDeltaSISO, or the SubBandCQIValueDeltaSCW, or SubBandCQIValueDeltaMCWLayer1 depending on the SFCH Report being SFCHSISO, SFCHSCW, or SFCHMCW respectively. (See Table 147.) See 8.7.5.4.15.3 for details regarding the computation of the SubBandCQIValue.

8.7.5.4.15.3 SubBandCQIValue

Depending on the SFCH Report Type, the SubBandCQIValueDelta can indicate the field SubBandCQIValueDeltaSISO in SFCHSISO report, the field SubBandCQIValueDeltaSCW in SFCHSCW report, or the field SubBandCQIValueDeltaMCWLayer1 in SFCHMCW report. Based on these values, the SubBandCQIValue is calculated as follows.

- If the SFCH Report Type is SFCHSISO, SubBandCQIValue = CQIValueSISO + SubBandCQIValueDeltaSISO.
- If the SFCH Report Type is SFCHSCW, SubBandCQIValue = MCQIValueSCW + SubBandCQIValueDeltaSCW.

- If the SFCH Report Type is SFCHMCW, $\text{SubBandCQIValueLayer1} = \text{MCQIValueMCWLayer1} + \text{SubBandCQIValueDeltaMCWLayer1}$.
- If the SFCH Report Type is SFCHMCW, $\text{SubBandCQIValueLayer2} = \text{MCQIValueMCWLayer2}$.
- If the SFCH Report Type is SFCHMCW, $\text{SubBandCQIValueLayer3} = \text{MCQIValueMCWLayer3}$.
- If the SFCH Report Type is SFCHMCW, $\text{SubBandCQIValueLayer4} = \text{MCQIValueMCWLayer4}$.

If r-bfch is enabled, the access terminal shall include the transmit processing gains corresponding to the BeamIndex as specified in the r-bfch, in the SubBandCQIValue. Precoding is defined in the Physical Layer Protocol. If binary unitary precoding is used with STTD with antenna selection indication, then SubBandCQIValueDelta also includes the effect of STTD over selected antennas

Based on an unrestricted observation interval, the access terminal shall compute the value of SubBandCQIValue_s for each subband s such that a packet transmitted by the access network to the access terminal over all of the hop-ports in subband s, sent at Reference Power Density using the specified packet format and the specified number of FL-PHY frames, terminating 1 PHY frame before the start of the PHY frame in which the R-SFCH is transmitted, would result in a packet error probability ≤ 0.01 . The CQI mappings are shown in Table 147. The access terminal shall report the value SubBandCQIValue_{SubBandIndex}.

8.7.5.4.16 Procedures for the r-bfch logical channel

The access terminal uses the r-bfch to transmit the beam index, and the CQI value offset necessary for SDMA transmission for the current FLSS. The CQI value shall be transmitted differentially with respect to the SubBandCQIValue if the r-sfch report is also being transmitted. SDMA is defined in the Physical Layer Protocol. A description of the beam code books, indexed by the BFCHBeamCodeBookIndex is also provided in the Physical Layer Protocol.

The BFCH reports are sent using the BFCHReport the details of which are in 8.7.5.4.16.2.

8.7.5.4.16.1 Reporting rules

The reporting rules for the r-bfch report are as follows:

- The r-bfch report may only be transmitted on OFDMA segments whose indices occur in the ODCCHReportIndex list.
- The access terminal shall send at least MinBFCHReportRate r-bfch reports over a set of eight consecutive ODCCH report opportunities. The access network shall ignore the r-bfch reports from the access terminal if MinBFCHReportRate is equal to the value '0' (this case will be referred to as r-bfch being disabled).

The access terminal shall not transmit the r-bfch after *AirLinkManagement.TunedAway* indication and until the *AirLinkManagement.TunedBack* indication.

8.7.5.4.16.2 BFCH report

Two modes shall be used for the BFCH Report. These will be specified by the parameter BFCHReportingMode. If BFCHReportingMode is SDMA, the broadband and subband CQI values shall be reported assuming SDMA, and the SDMADeltaCQI field in the r-bfch report shall carry the amount of CQI gain in the absence of SDMA. If BFCHReportingMode is noSDMA, the broadband and subband CQI values shall be reported assuming no SDMA, and the SDMADeltaCQI field in the r-bfch report shall include the amount of CQI loss in the presence of SDMA. In either case, the format of the BFCH Report will be the same. The interpretation will depend on the BFCHReportingMode.

The r-bfch reports associated with r-cqich and r-mqich are shown in Table 149. The access terminal shall determine that CQIMode is SISO if the MIMOMode is MCW and the MaxMCWLayers in the FTC-MAC is set to 1. It shall determine that the CQIMode is SCW if the MIMOMode is SCW and it shall determine that the CQIMode is MCW if the MIMOMode is MCW and MaxMCWLayers > 1. This will also indicate the BFCH Report Type as shown in Table 149.

Table 149—BFCH Report Types

	CQIMode		
	SISO	SCW	MCW
BFCH Report	BFCHSISO	BFCHSCW	BFCHMIMO

The format for BFCHReport is shown in Table 150.

Table 150—Format for BFCHReport

Field	Length (bits)
BeamIndex	6
SDMADeltaCQI	2

BeamIndex

The interpretation of the non-zero values of the BeamIndex field is as described in 8.7.5.4.16.3. The access terminal can use the value 0 to indicate no preferred beam. In this case, the access terminal shall report non-precoded CQI, as defined in 8.7.5.4.12, on the relevant logical channel (r-sfch or r-mqich depending on whether r-sfch reports are enabled or not), or STTD CQI value when applicable, see 8.7.5.4.13.2.

SDMADeltaCQI

If r-sfch is enabled and the R-ODCCH report has a header value of '000', the SDMADeltaCQI indicates the integer offset that is to be added/subtracted (depending on the BFCHReportingMode) from the SubBandCQIValue field of the SFCH reports, to obtain the SDMACQIValue. See 8.7.5.4.16.4 for details on SDMACQIValue.

If r-sfch is enabled or the R-ODCCH report has a header value of '011' or '101', the SDMADeltaCQI indicates the integer offset that is to be added/subtracted (depending on the BFCHReportingMode) from the appropriate CQIValue field of the CQI reports, to obtain the SDMACQIValue. See 8.7.5.4.16.4 for details on SDMACQIValue.

8.7.5.4.16.3 Interpretation of BeamIndex Field

The interpretation of the BeamIndex Field depends on the BFCHBeamCodeBookIndex, which is specified in the OMP, and the BFCHCodebookType field of the codebook specified by the BFCHBeamCodebookIndex parameter. The BFCHCodebookType can take two values: '0' (for “KnockDown”) and '1' (for “ReadyMade”). Let N be the number of transmit antennas. The BeamIndex Field shall be read from left to right as MSB to LSB according to the following rules:

- If BFCHCodebookType = 0 and BFCHBeamCodebookIndex = 0, this corresponds to Binary Unitary codebook. The N LSBs of the BeamIndex Field are a Vector Bitmap (VBM) that indicate which antennas are selected. If the access terminal does not support STTD with antenna selection, i.e., STTDAntennaSelectionSupported = 0, the remaining (6 – N) bits of the BeamIndex field are

reserved. If the access terminal supports STTD with antenna selection, i.e., STTDAntennaSelectionSupported = 1, if $N < 5$ and $N > 2$ and if two antennas are selected, the $(N + 1)^{\text{th}}$ LSB of the BeamIndex Field indicates if STTD is present. The remaining $(5 - N)$ bits are Reserved.

- b) If BFCHCodebookType = 0 and BFCHBeamCodebookIndex = 1, i.e., Fourier based codebook:
 - 1) If $N < 6$, a vector of N LSBs of BeamIndex field specifies Vector Bitmap (VBM) of length N , which indicates a selected set of columns of a precoding matrix. If the number of precoding matrices defined by the precoding codebook with index BFCHBeamCodebookIndex is 2 then the $(N + 1)^{\text{th}}$ LSB of BeamIndex field contains the Preferred Matrix Index (PMI) that specifies which of the two matrices is used. If the number of precoding matrices defined by the precoding codebook with index BFCHBeamCodebookIndex is 1 then the $(N + 1)^{\text{th}}$ LSB of BeamIndex field is reserved. The remaining $(5 - N)$ bits of BeamIndex field are reserved.
 - 2) If $N = 6$, BeamIndex field specifies Vector Bitmap (VBM) of length 6, which indicates a selected set of columns of a precoding matrix defined by the precoding codebook with index BFCHBeamCodebookIndex.
- c) If BFCHCodebookType = 1, i.e., ReadyMade codebook: BeamIndex field specifies Precoder Index (PCI) of the selected precoding matrix defined by the precoding codebook with index BFCHBeamCodebookIndex.

8.7.5.4.16.4 SDMACQIValue

The SDMACQIValue is to be used by the access network for SDMA transmission.

In this subclause, the parameters used to compute SDMACQIValue are to be interpreted as integers. If SDMACQIValue is computed to be a number smaller than 0, the SDMACQIValue shall be made equal to 0. SDMACQIValue shall be interpreted as a 5-bit CQI value if the CQIMode is SCW and a 4-bit value otherwise. The CQI mappings to be used for SDMACQIValue are shown in Table 144.

If BFCHReportingMode is SDMA, the precoded CQI values reported on r-mqich or r-sfch include SDMA loss, and are SDMACQIValues. In this case, the NonSDMACQIValues will be computed using SDMADeltaCQI field of the r-bfch report as described below. If BFCHReportingMode is noSDMA, the precoded CQI values reported on r-mqich or r-sfch do not include SDMA loss, and are NonSDMACQIValues. In this case, the SDMACQIValues will be computed using SDMADeltaCQI field of the r-bfch report as described below.

If the MinSFCHReportRate is 0 or if the R-ODCCH report has a header value of '011' or '101', the SDMACQIValue is computed as follows. The SDMADeltaCQI refers to the broadband CQI in this case.

If the CQIMode is SISO,

- If BFCHReportingMode is SDMA, then $\text{NonSDMACQIValue} = \text{CQIValueSISO} + \text{SDMADeltaCQI}$
- If BFCHReportingMode is noSDMA, then $\text{SDMACQIValue} = \text{CQIValueSISO} - \text{SDMADeltaCQI}$

where CQIValueSISO is a field in the CQICHCTRL report of the r-cqich, described in 8.7.5.4.11.2.

If the CQIMode is SCW,

- If BFCHReportingMode is SDMA, then $\text{NonSDMACQIValue} = \text{MCQIValueSCW} + \text{SDMADeltaCQI}$
- If BFCHReportingMode is noSDMA, then $\text{SDMACQIValue} = \text{MCQIValueSCW} - \text{SDMADeltaCQI}$

where MCQIValueSCW is a field in the MQICHSCW report of the r-mqich, described in 8.7.5.4.13.2.

If the CQIMode is MCW,

- If BFCHReportingMode is SDMA, then the NonSDMACQIValue for each MIMO layer k defined as $\text{NonSDMACQIValue}_k = \text{MCQIValueMCWLayer}_k + \text{SDMA}\Delta\text{CQI}$
- If BFCHReportingMode is noSDMA, then the SDMACQIValue for each MIMO layer k defined as $\text{SDMACQIValue}_k = \text{MCQIValueMCWLayer}_k - \text{SDMA}\Delta\text{CQI}$

where CQIValueMCWLayer k is a field in the MQICHMCW report of the r-mqich, described in 8.7.5.4.13.2.

If the MinSFCHReportRate is not 0 and the R-ODCCH report has a header value of '000', the SDMACQIValue is computed as follows.

If the CQIMode is SISO,

- If BFCHReportingMode is SDMA, then $\text{NonSDMACQIValue} = \text{CQIValueSISO} + \text{SubBandCQIValueDeltaSISO} + \text{SDMA}\Delta\text{CQI}$
- If BFCHReportingMode is noSDMA, then $\text{SDMACQIValue} = \text{CQIValueSISO} + \text{SubBandCQIValueDeltaSISO} - \text{SDMA}\Delta\text{CQI}$

where SubBandCQIValueDeltaSISO is a field in the SFCHSISO report of the r-sfch, described in 8.7.5.4.15.2 and CQIValueSISO is a field in the CQICHCTRL report of the r-cqich, described in 8.7.5.4.11.2.

If the CQIMode is SCW,

- If BFCHReportingMode is SDMA, then $\text{NonSDMACQIValue} = \text{MCQIValueSCW} + \text{SubBandCQIValueDeltaSCW} + \text{SDMA}\Delta\text{CQI}$
- If BFCHReportingMode is noSDMA, then $\text{SDMACQIValue} = \text{MCQIValueSCW} + \text{SubBandCQIValueDeltaSCW} - \text{SDMA}\Delta\text{CQI}$

where SubBandCQIValueDeltaSCW is a field in the SFCHSCW report of the r-sfch, described in 8.7.5.4.15.2 and MCQIValueSCW is a field in the MQICHSCW Report of the r-mqich, described in 8.7.5.4.13.2.

If the CQIMode is MCW,

- If BFCHReportingMode is SDMA, then the NonSDMACQIValue for MIMO layer 1 is defined as $\text{NonSDMACQIValue} = \text{MCQIValueMCWLayer1} + \text{SubBandCQIValueDeltaMCWLayer1} + \text{SDMA}\Delta\text{CQI}$
- If BFCHReportingMode is noSDMA, then the SDMACQIValue for MIMO layer 1 is defined as $\text{SDMACQIValue} = \text{MCQIValueMCWLayer1} + \text{SubBandCQIValueDeltaMCWLayer1} - \text{SDMA}\Delta\text{CQI}$

where SubBandCQIValueDeltaMCWLayer1 is a field in the SFCHMCW report of the r-sfch, described in 8.7.5.4.15.2 and MCQIValueMCWLayer1 is a field in the MQICHMCW Report of the r-mqich, described in 8.7.5.4.13.2.

If the CQIMode is MCW,

- If BFCHReportingMode is SDMA, then the NonSDMACQIValue for MIMO layer k , where k is 2 or 3 or 4, is defined as $\text{NonSDMACQIValue}_k = \text{MCQIValueMCWLayer}_k + \text{SDMA}\Delta\text{CQI}$
- If BFCHReportingMode is noSDMA, then the SDMACQIValue for MIMO layer k , where k is 2 or 3 or 4, is defined as $\text{SDMACQIValue}_k = \text{MCQIValueMCWLayer}_k - \text{SDMA}\Delta\text{CQI}$

where MCQIValueMCWLayer k is a field in the MQICHMCW Report of the r-mqich, described in 8.7.5.4.13.2.

In computing the SDMACQIValue, the access terminal shall assume an interfering terminal with fully overlapping assignment in the worst interfering cluster of the codebook specified by BFCHBeamCodebookIndex. The worst interfering cluster is defined as a cluster that, according to the OverlapMap of the codebook, is allowed to overlap with the cluster from which the precoding matrix specified by the BeamIndex parameter is selected, and contains a precoding matrix that, if used for an interfering access terminal, results in the lowest SDMACQIValue for the desired access terminal. The access terminal shall use this lowest SDMACQIValue to compute the SDMADeltaCQI, according to the following rules:

- If r-sfch is disabled and the R-ODCCH report has a header value of '000', the access terminal shall report the lowest tabulated SDMADeltaCQI based on an unrestricted observation interval, such that the packet transmitted by the access network to the access terminal over all of the hop-ports referred by the SubBandIndex field of the SFCH report, sent at Reference Power Density using the specified packet format and the specified number of FL-PHY frames corresponding to SDMACQIValue, terminating 1 PHY frame before the start of the PHY frame in which the r-bfch is transmitted, would result in a packet error probability ≤ 0.01 .
- If r-sfch is disabled or the R-ODCCH report has a header value of '011' or '101', the access terminal shall report the lowest tabulated SDMADeltaCQI based on an unrestricted observation interval, such that the packet transmitted by the access network to the access terminal over all of the hop-ports used in the computation of the broadband CQI, sent at Reference Power Density using the specified packet format and the specified number of FL-PHY frames corresponding to SDMACQIValue, terminating 1 PHY frame before the start of the PHY frame in which the r-bfch is transmitted, would result in a packet error probability ≤ 0.01 .

8.7.5.4.17 Procedures for the r-reqch logical channel

This 6-bit logical channel may be carried on the R-CDCCH or the R-ODCCH. It is used by the access terminal to request resources based on the QoS required and the status of the buffers at the access terminal.

8.7.5.4.17.1 Reporting rules

The access terminal shall not transmit the r-reqch after *AirLinkManagement.TunedAway* indication and until *AirLinkManagement.TunedBack* indication. The reporting rules for r-reqch are as follows:

- If the r-reqch is being carried on the R-CDCCH, the RCC MAC Protocol shall provide the Physical Layer Protocol with the P_{REQCH} , PilotPN, and MACID. The PilotID and MACID correspond to the target sector. The P_{REQCH} is determined according to the rules in 8.7.5.4.16.3. The r-reqch may be transmitted on any subsegment assigned for primary R-CDCCH transmission of the access terminal and specified by CDCCHSubsegmentIndex, CDCCHPeriod, and CDCCHPhase list.
- The P_{REQCH} is determined according to the rules in 8.7.5.4.6.3. The access terminal may use an additional REQQoSPowerBoost dB of power on this channel if the request is being made for a stream with StreamNNRequestPowerBoostAllowed attribute equal to one.
- For r-reqch transmissions to the FLSS on the R-ODCCH, and the access terminal may use any ODCCHReportIndex that is assigned to the access terminal for R-ODCCH transmissions to the FLSS.

8.7.5.4.17.1.1 Transmitting r-reqch on the R-CDCCH when R-ODCCH resources assigned

When the access terminal has assigned resources on R-ODCCH, the access terminal shall not send r-reqch on the R-CDCCH if any of the following conditions is true:

- There is an available R-ODCCH opportunity within the next eight PHY frames
- The access terminal transmitted r-reqch in the last MinRequestInterval PHY frames

8.7.5.4.17.1.2 Transmitting r-reqch on the R-CDCCH when R-ODCCH resources not assigned

When the access terminal has no assigned resources on the R-ODCCH, the access terminal shall send r-reqch on the R-CDCCH according to the following rules.

While an InBand request timer is active (as indicated by the Reverse Traffic Channel MAC), or if the access terminal had sent a r-reqch transmission within the last $\text{MinRequestInterval } N_{\text{Backoff}} \times \text{PHY frames}$, the access terminal shall not send r-reqch on the R-CDCCH unless at least one of the following exception rules are satisfied:

- The Stream/NNPriority of the r-reqch transmission is higher than the Stream/NNPriority of the stream with highest priority at the time the last request was sent
- The r-reqch transmission is to indicate a latency requirement of 20 ms or less, and the last r-reqch transmission did not signal a latency requirement of less than 20 ms.

Here, N_{Backoff} is selected as follows:

- 1) Set $p = \text{REQBackoffFactor}^M$
- 2) If $p = 0$, set $n = 0$. Otherwise, generate a random variable that takes value $n \geq 0$ with probability $p(1 - p)^n$
- 3) Calculate n_{max} as the smallest value of n for which $p(1 - p)^n$ is less than 0.01
- 4) Set N_{Backoff} to $\min(n, n_{\text{max}})$

Here M is a counter that is maintained according to the following rules:

- a) It is reset to 0 each time
 - The InBand request timer is reset.
 - The InBand request timer expires.
 - The access terminal receives a new RL assignment.
 - The access terminal has not received an RL assignment for a time interval larger than $\text{REQBackoffResetTime}$.
 - The access terminal determines that it needs to send an r-reqch report based on the first exception rule listed above.
 - There is a change in DRLSS.
- b) It is incremented by one each time an r-reqch report is transmitted on R-CDCCH.

8.7.5.4.17.2 REQCHReport

The six bits of the r-reqch are read from the left (as the MSB) to the right (as the LSB). Then, follow the rules below.

To compute the REQCHReport, the access terminal first computes the quantity BacklogBytes as the smallest MAC Layer assignment required to transmit the data buffered at the access terminal. For example, if the access terminal has 40 bytes of buffered data (including all upper layer and PCP headers), and 1 byte of InBand signaling, and the CRC is 3 bytes, then BacklogBytes is equal to 44.

The access terminal maps BacklogBytes to BacklogLevel using a spectral efficiency-dependent mapping. During PHY frame n , define SE as the spectral efficiency of the last reverse link assignment received before RL PHY frame n_a , where $n_a = n - 4$ if DuplexingMode = FDD or DuplexingMode = TDD44 and $n_a = n - 3$ if DuplexingMode = TDD63. If the access terminal has not received any reverse link assignment from this sector, or is requesting handoff to this sector, the SE shall be assumed to be 0.5.

The access terminal shall not send a REQCHReport for which it is unable to construct a data packet under the constraints imposed by the Packet Consolidation Protocol. For example, if the Packet Consolidation Protocol allows 100 bytes to be transmitted on the Reverse Traffic Channel for a given stream (even when there are say 1500 bytes for that stream in the buffer), the access terminal shall not request a value larger than 100 bytes.

8.7.5.4.17.2.1 REQ format for values less than 48

If the first three bits give an integer between 0 to 5, then the first three bits refer to the BacklogLevel and the next three bits refer to the BacklogType.

8.7.5.4.17.2.1.1 Interpretation of first three bits

For $SE > 1$ (except SE corresponding to Packet Format 5), the access terminal shall use the mapping described in Table 151. If $SE \leq 1$, or corresponds to Packet Format 5, the access terminal shall use the mapping described in Table 152. When using these tables, the access terminal may round up, i.e., to request BacklogBytes = 40, it may use the value corresponding to BacklogBytes = 64. For values larger than the last value in the table, the access terminal may use the largest BacklogLevel.

Table 151—BacklogLevel and BacklogBytes for $SE > 1$ (except Packet Format 5)

BacklogLevel	BacklogBytes
0	$34 \times SE$
1	$64 \times SE$
2	$128 \times SE$
3	$256 \times SE$
4	$512 \times SE$
5	$> 512 \times SE$

The relationship between the three bits indicating the BacklogLevel and the number of Base Tiles that the access terminal wishes to request is given in Table 152. The access terminal may always round up the number of Base Tiles it requests. For example, if it wants to request three Base Tiles, the actual request it transmits will be for four Base Tiles.

Table 152—Relation between BacklogLevel, BacklogBytes, and number of Base Tiles requested for $SE \leq 1$ or Packet Format 5

BacklogLevel	BacklogBytes
0	34
1	64
2	128
3	256
4	512
5	>512

8.7.5.4.17.2.1.2 Interpretation of last three bits

The next three bits are used to indicate the BacklogType. There are three modes for reporting this. The mode is specified by the REQConfig parameter, carried in the OMP. This parameter takes values 1, 2, and 3. Depending on the value of this parameter, the BacklogType is specified either in terms of latency deadline or QoSClass. The latency deadline is equal to the time until packet expiry, based on the time of arrival of the packet, and the assigned TransmitAbortTimerRev public data of the Radio Link Protocol corresponding to the Stream NN for the stream to which that the packet belongs.

The three modes are explained as follows.

If REQConfig = 1 (see Table 153), BacklogType is specified in terms of eight possible Cfg1QoSClass values. The Cfg1QoSClass value for Stream NN is indicated in the Stream NN ReqChannelCfg1QoSClass attribute.

Table 153—Interpretation of BacklogType when REQConfig =1

BacklogType	Interpretation
0-7	Cfg1QoSClass

If REQConfig = 2 (see Table 154), BacklogType is specified in terms of either four latency deadlines or four possible Cfg2QoSClass values. If a particular stream is not assigned a time limit (i.e., TransmitAbortTimer for the stream is 0), then the access terminal shall request for that flow by using BacklogType corresponding to the assigned Cfg2QoSClass. The Cfg2QoSClass value for Stream NN is indicated in the Stream NN ReqChannelCfg2QoSClass attribute.

Table 154—Interpretation of BacklogType when REQConfig =2

BacklogType		Interpretation
0-3		Cfg2QoSClass
4	Latency Deadline (ms)	20
5		40
6		80
7		120

If REQConfig = 3 (see Table 155), BacklogType is specified solely in terms of latency deadline. For flows where a latency deadline is not specified, the access terminal shall request using the total backlog method, described in 8.7.5.4.17.2.2.

Table 155—Interpretation of BacklogType when REQConfig = 3

BacklogType	Latency deadline (ms)
0	20
1	40
2	60
3	80
4	100

Table 155—Interpretation of BacklogType when REQConfig = 3 (continued)

BacklogType	Latency deadline (ms)
5	120
6	160
7	200

8.7.5.4.17.2.2 REQ format for values 48 and larger

If the first three bits give an integer that is 6 or 7, then the 6-bit value on the r-reqch is interpreted as total backlog in bytes, as defined by Table 156. In the table, SE is the spectral efficiency of the last reverse link assignment received before PHY frame $n - 4$.

Table 156—Interpretation of r-reqch for values greater than 48 (notation: $k = 1024$)

r-reqch value	BacklogBytes
110000	$64 \times \text{SE}$
110001	$128 \times \text{SE}$
110010	$256 \times \text{SE}$
110011	$384 \times \text{SE}$
110100	$512 \times \text{SE}$
110101	$1024 \times \text{SE}$
110110	$1536 \times \text{SE}$
110111	$2k \times \text{SE}$
111000	$4k \times \text{SE}$
111001	$6k \times \text{SE}$
111010	$8k \times \text{SE}$
111011	$12k \times \text{SE}$
111100	$16k \times \text{SE}$
111101	$32k \times \text{SE}$
111110	$48k \times \text{SE}$
111111	$64k \times \text{SE}$

8.7.5.4.18 Procedures for the r-pahch logical channel

This is a 6-bit logical channel that is carried on the R-CDCCH physical channel. It carries the maximum achievable CoT value over the entire band.

The CoT is defined as the average carrier power divided by the average thermal power in the band in dB, that is $\text{CoT} = 10\log_{10}(\text{Carrier Power}/\text{Thermal Power})$. This shall be computed using the pilot CoT feedback on F-PQICH. The quantity transmitted is

$$P_{\text{Max}} - P_{\text{PICH}} - 10\log_{10}(N_{\text{FFT}}) + p\text{CoT}_{\text{RLSS}}$$

where P_{Max} is the maximum power at which access terminal can make RL transmissions for an assignment that spans the entire bandwidth and N_{FFT} is given by the TotalNumSubcarriers field of the OMP.

8.7.5.4.18.1 Reporting rules

The r-pahch is transmitted only when the value to be transmitted is substantially different from the previous report. The access terminal may choose to transmit the r-pahch to the RLSS on the CDMA subsegment specified by CDCCHSubsegmentIndex, CDCCHPeriod, and CDCCHPhase, subject to the following restrictions:

- The InBand PAH Timer of the RTC MAC Protocol is not active.
- The change in the r-pahch report relative to the previous r-pahch report is at least MinPAHChange.
- The number of r-pahch reports in any set of 32 consecutive R-PICH opportunities $32k$, $32k + 1$, ..., $32k + 31$ does not exceed MaxPAHCHReportRate.

8.7.5.4.18.2 PAHCH report

The format for the r-pahch report is as follows in Table 157.

Table 157—Format for PAHCHReport

Field	Length (in bits)
PAHCHValue	6

PAHCHValue Indicates PA Headroom value in dB. See 8.7.5.4.18.3 for details.

8.7.5.4.18.3 PAHCHValue

The 6 bits in the PAHCHValue field of the PAHCHReport indicate values in the range -26 dB to 26 dB as follows. The access terminal shall report the value from Table 158 that is closest to the observed value.

Table 158—PAHCHValue in dB

PAHCHValue	Value in dB
0–31	-26 to -10.5 in steps of 0.5
32–57	-10 to 15 in steps of 1
58–63	16 to 26 in steps of 2

8.7.5.4.19 Procedures for the r-psdch logical channel

This is a 4-bit logical channel that is carried on the R-CDCCH physical channel. It carries information about the minimum ChanDiff value computed by the access terminal. The notion of ChanDiff is defined in 8.7.5.4.19.1.

8.7.5.4.19.1 Reporting rules

The r-psdch is transmitted only where the minimum ChanDiff value changes substantially from the previous report. The access terminal may choose to transmit the r-psdch to the RLSS on the CDMA subsegment specified by CDCCHSubsegmentIndex, CDCCHPeriod, and CDCCHPhase, subject to the following restrictions:

- The change in the r-psdch report relative to the previous r-psdch report is at least MinChanDiffChange.
- The number of reports in any set of 32 consecutive R-PICH opportunities $32k, 32k + 1, \dots, 32k + 31$ does not exceed MaxPSDCHReportRate.

8.7.5.4.19.2 PSDCH Report

The format for the r-psdch report is as follows in Table 159.

Table 159—Format for PSDCHReport

Field	Length (in bits)
PSDCHValue	4

PSDCHValue Indicates minimum ChanDiff value in dB. See 8.7.5.4.19.3 for details

8.7.5.4.19.3 PSDCHValue

The 4 bits in the PSDCHValue field of the PSDCHReport indicate values in the range -5 dB to 15 dB as follows in Table 160. The access terminal shall report the value from this table that is closest to the observed value.

Table 160—PSDCHValue in dB

PSDCHValue	Value in dB
0–10	-5 to 5 in steps of 1
11–15	7 to 15 in steps of 2

8.7.5.4.20 Procedures for the R-ACKCH Physical Layer channel

8.7.5.4.20.1 Terms

The following definitions will be used in this subclause, and apply to a specified RL PHY frame.

RACKNodeIndex Each usable node on the FL channel tree, as well as each acknowledgeable F-SCCH block, is associated with a RACKNodeIndex. The assignment of the RACK node indices is explained in 8.7.5.4.20.2. The RACKNodeIndex is common to all access terminals.

NumRACKNodesDCH This is the total number of R-ACK node indices used for the data channels. Thus, the RACKNodeIndex parameter for data channels takes values from 0 to $\text{NumRACKNodesDCH} - 1$.

NumRACKNodes This is the total number of RACK node indices used for data channels as well as for acknowledging assignments. The RACKNodeIndex parameter for F-SCCH blocks takes values from NumRACKNodesDCH to $\text{NumRACKNodes} - 1$.

access terminal assignment

This is the set of base nodes assigned to the access terminal by the FTC MAC.

SubtreeIndex	This is the index of the subtree from which the base nodes have been assigned. See 8.6.5.6.
PDCABResourceSharingEnabled	This bit is 1 if PDCAB is enabled; otherwise, it is 0. See 8.5.5.4.1.1.3.4 for more details.
PDCABSupported	This bit is set to '1' if the PDCAB feature is supported by the access terminal.
RACKBandwidthFactor	If r is the RACKBandwidthFactor, then every r^{th} base node gets indexed.
NumAckableLABs	This is the number of LinkAssignmentBlocks (LABs) that can be acknowledged.
MaxNumSharedLABs	This is the number of LinkAssignmentBlocks that are decoded by all access terminals.
MaxNumQPSKLABs	This is the maximum number of QPSK LABs that are transmitted in each FL PHY frame, and is computed by the Physical Layer Protocol.

8.7.5.4.20.2 Determining the RACK node indices

Selected base nodes of the forward link channel tree are assigned an RACKNodeIndex according to the following procedure:

- a) Initialize a base node counter b to the smallest base node index in subtree 0. Initialize an RACKNodeIndex counter k to 0.
- b) If base node b only comprises of usable hop-ports (as defined by the Physical Layer Protocol) and both PDCABResourceSharingEnabled and PDCABSupported are set to '1', or, if base node b only comprises of usable hop-ports, either PDCABResourceSharingEnabled or PDCABSupported is set to '0', and base node b is not assigned to the Forward Link Control Segment (as specified by the Physical Layer Protocol), then
 - 1) The RACKNodeIndex of this base node on all subtrees shall be k .
 - 2) Increment k by 1.
- c) Increment b by RACKBandwidthFactor.
- d) If b is smaller than the largest base node index in subtree 0, go to step b).
- e) Set NumRACKNodesDCH = k .

8.7.5.4.20.3 ACK reporting rules

The access terminal shall not transmit the R-ACKCH after *AirLinkManagement.TunedAway* indication and until the *AirLinkManagement.TunedBack* indication.

The relationship between the FL PHY frame on which a packet is transmitted, and the RL PHY frame carrying the ACK is provided in 8.1.5. If DuplexingMode = FDD or DuplexingMode = TDD44, each RL PHY frame carries acknowledgements for a single FL PHY frame, while if DuplexingMode = TDD63, each RL PHY frame carries acknowledgements for a single pair of successive FL PHY frames.

The RACKVal computation and the rules for which RACKNodeIndex to transmit that RACKVal on are described in 8.7.5.4.20.3.1. The P_{ACKCH} is determined according to the rules in 8.7.5.4.6.3.

This protocol shall provide the Physical Layer with NumRACKNodes, P_{ACKCH} and RACKNodeIndex and RACKVal, for each acknowledgement sent on the RL PHY frame.

8.7.5.4.20.3.1 RACKVal and RACKBaseNodeIndex Computation

The rules for computing the RACKVal and the RACKNodeIndex to transmit it on are stated next. The access network shall ensure that the access terminal assignments allow for the following constraints to be met.

Let m denote the SubtreeIndex of the forward link access terminal assignment. Let N_{ACKABLE} denote the number of base nodes within the access terminal assignment that are assigned a RACKNodeIndex and used by the access terminal in the last FL PHY frame of the HARQ transmission being acknowledged. Refer to these base-nodes as ackable nodes 0, 1, ..., $N - 1$, where the indexing starts from the base-node with the smallest NodeID and successive indices are assigned to base-nodes in increasing order of their NodeIDs.²³

- If DuplexingMode = TDD63, a MAC packet that is split in two F-DCH packets that are then encoded, transmitted, and decoded separately over a pair of successive FL PHY frames is said to pass the CRC if the CRC passes for both F-DCH packets that are decoded individually.
- For the SISO or MIMO-SCW acknowledgement, set $M = 1$. If the FL transmission of the Lower MAC packet passes the CRC as described by the Physical Layer Protocol, the RACKVal is set to 1; otherwise, it is set to 0. The RACKNodeIndex on which this RACKVal is transmitted is given by the RACKNodeIndex of the base-node with the ackable node index of $\min(m, N_{\text{ACKABLE}} - 1)$.
- For the MIMO-MCW acknowledgement, let M be the number of layers being used. For each layer k , the RACKVal is set to 1 if the Lower MAC packet on that layer passes the CRC as described by the Physical Layer Protocol; otherwise, it is set to 0. The RACKNodeIndex on which this RACKVal is transmitted is given by the RACKNodeIndex of the base-node with the ackable node index of $\min(m, N_{\text{ACKABLE}} - M) + k - 1$. This is done for k going from 0 to M .

The access network shall ensure that the access terminal assignments are such that the above rules assign different RACKNodeIndex values to different access terminals on a given PHY frame.

8.7.5.4.20.4 Acknowledging LinkAssignmentBlocks (LABs)

In addition to carrying acknowledgements for F-DCH packets, the R-ACKCH also carries acknowledgements for forward and reverse link assignments received on the F-SCCH. The RCC MAC sends an acknowledgement for an assignment when issued a *ReverseControlChannelMAC.ACKAssignment* from the FTC or RTC MAC Protocol. The FTC or RTC MAC Protocol provides the index of the LAB to be acknowledged, as well as the target sector, along with this indication. For a LAB index k , define a RACKNodeIndex according to the following procedure:

- If $k < \text{NumAckableLABs}$ and $k < \text{MaxNumSharedLABs}$, then set $\text{RACKNodeIndex} = \text{NumRACKNodesDCH} + k$
- If $\text{NumAckableLABs} > \text{MaxNumSharedLABs}$, then the LAB with index $k = \text{MaxNumSharedLABs} + i \times \text{floor}((\text{MaxNumQPSKLABs} - \text{MaxNumSharedLABs}) / (\text{NumAckableLABs} - \text{MaxNumSharedLABs}))$ is assigned $\text{RACKNodeIndex} = \text{NumRACKNodesDCH} + \text{MaxNumSharedLABs} + i$.
- The remaining LAB indices are not assigned any RACKNodeIndex
- The access terminal shall set $\text{NumRACKNodes} = \text{NumRACKNodesDCH} + \text{NumAckableLABs}$.

For the LAB indices received from the AC, FTC, or RTC MAC Protocol along with the *ReverseControlChannelMAC.ACKAssignment* indication that are assigned a RACKNodeIndex and that are to be acknowledged in the same RL PHY frame, a RACKVal = 1 shall be sent on this RACKNodeIndex to

²³Note that a group resource assignment can occupy different base nodes in different FL PHY frames of the same HARQ transmission. With other types of FL assignments, the same set of base nodes is occupied in all the FL PHY frames of the same HARQ transmission. In the latter case, the value of T can be computed based on any of the FL PHY frames within the same HARQ transmission.

the target sector, if the corresponding LAB index is within the lowest MaxNumSCCHacks LAB indices that satisfy the previous conditions. No action shall be carried out for the other F-SCCH blocks.

8.7.5.4.21 Procedures for MIMOResponse message

If MIMO feedback is disabled, the access terminal shall transmit the MIMOResponse message to the access network if the last MIMOResponse message was transmitted more than MIMOResponseMessageTimer time ago, where MIMOResponseMessageTimer is a configuration attribute of this protocol, and following condition is met:

$$\frac{SE_{MIMO}}{SE_{CURRENT}} \geq MIMOResponseMessageThreshold$$

where

- SE_{MIMO} is the anticipated spectral efficiency if the access network employs a MIMO scheme
- $SE_{CURRENT}$ is the anticipated spectral efficiency if the access network maintains the current antenna configuration
- MIMOResponseMessageThreshold is a protocol attribute

Otherwise, the access terminal shall not transmit the MIMOResponse message.

8.7.5.5 Parameter management and message processing

The access terminal shall maintain a parameter Override for each sector in the Active Set. Upon addition of a sector to the Active Set, the access terminal shall initialize Override = NULL. If the access terminal receives a ControlPowerControlOverride message for a sector, it shall set Override = NON-NULL for that sector.

If Override = NULL, the access terminal shall set the parameters PowerControlStepSize, CQITargetCtoI, REQTargetCtoI, PAHTargetCtoI, PSDTargetCtoI, ODCTargetCtoI, ACKTargetCtoI, MinPAHChange, and MinChanDiffChange to PowerControlStepSizeAttribute, CQITargetCtoIAttribute, REQTargetCtoIAttribute, PAHTargetCtoIAttribute, PSDTargetCtoIAttribute, ODCTargetCtoIAttribute, ACKTargetCtoIAttribute, MinPAHChangeAttribute, and MinChanDiffChangeAttribute, respectively.

If Override = NON-NULL, the access terminal shall set PowerControlStepSize, CQITargetCtoI, REQTargetCtoI, PAHTargetCtoI, PSDTargetCtoI, ODCTargetCtoI, ACKTargetCtoI, MinPAHChange, and MinChanDiffChange to the values received in the last received override message.

8.7.6 Message formats

8.7.6.1 ControlPowerControlOverride

This message may be used by the access network to reset some power control related attributes at the access terminal for a particular pilot in the active set (see Table 161).

Table 161—ControlPowerControlOverride message format

Field	Length (bits)
MessageID	8
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
PowerControlStepSizeOverride	3
CQITargetCtoIOverride	3
REQTargetCtoIOverride	3
PAHTargetCtoIOverride	3
PSDTargetCtoIOverride	3
ODCTargetCtoIOverride	3
ACKTargetCtoIOverride	3
MinPAHChangeOverride	3
MinChanDiffChangeOverride	3
Padding	Variable

MessageID	The access network shall set this field to 0x00.
PilotID	The access network shall set this field to the PilotID corresponding to the sector for which this message is valid.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the pilot to which this message refers. This field shall be omitted otherwise.
PowerControlStepSizeOverride	The new value for PowerControlStepSize and interpreted similarly to the corresponding configuration attribute.
CQITargetCtoIOverride	The new value for CQITargetCtoI and interpreted similarly to the corresponding configuration attribute.
REQTargetCtoIOverride	The new value for REQTargetCtoI and interpreted similarly to the corresponding configuration attribute.

PAHtargetCtoIOverride	The new value for PAHtargetCtoI and interpreted similarly to the corresponding configuration attribute.
PSDTargetCtoIOverride	The new value for PSDTargetCtoI and interpreted similarly to the corresponding configuration attribute.
ODCTargetCtoIOverride	The new value for ODCTargetCtoI and interpreted similarly to the corresponding configuration attribute.
ACKTargetCtoIOverride	The new value for ACKTargetCtoI and interpreted similarly to the corresponding configuration attribute.
MinPAHChangeOverride	The new value for MinPAHChange and interpreted similarly to the corresponding configuration attribute.
MinChanDiffChangeOverride	The new value for MinChanDiffChange and interpreted similarly to the corresponding configuration attribute.
Padding	The length of this field shall be such that this message is octet aligned. The sender shall set this field to zero. The receiver shall ignore this field.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key is available

8.7.6.2 MIMOResponse

If MIMO feedback is disabled, the access terminal sends the MIMOResponse message to request the access network to enable the MIMO feedback channel (see Table 162).

Table 162—MIMOResponse message format

Field	Length (bits)
MessageID	8
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
MIMOToCurrentRatio	3
Padding	Variable

MessageID	The access terminal shall set this field to 0x01.
PilotID	The access terminal shall set this field to PilotID of the sector for which it is requesting to enable the MIMO feedback channels.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.

ChannelBand This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.

MIMOToCurrentRatio The access terminal shall set this field to the largest value which is less than or equal to the ratio of the anticipated MIMO spectral efficiency to the anticipated spectral efficiency using the current antenna configuration. The encoding of this field is specified in Table 163.

Table 163—Ratio of MIMO to current

Field value (binary)	Ratio of MIMO to current
'000'	1.2
'001'	1.6
'010'	2.0
'011'	2.4
'100'	2.8
'101'	3.2
'110'	3.6
'111'	Reserved

Padding The length of this field shall be such that this message is octet aligned. The access terminal shall set this field to 0. The access network shall ignore this field.

Channels	RTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key is available

8.7.7 Configuration attributes

8.7.7.1 Simple attributes

The simple configuration attributes are listed in Table 164. The access network and the access terminal shall use the default values that are typed in bold italics.

Table 164—Simple configuration attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	IntraCellCQISwitchTimer	Soft Allowed/Static	0x00	The access network does not provide any suggestion to the access terminal about intra-cell handoff requirements.
			0x01–0xff	The access terminal should not perform two Intra-Cell FL or RL switches within IntraCellCQISwitchTimer. This field is in units of 25 ms.

Table 164—Simple configuration attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0001	InterCellCQISwitchTimer	Soft Allowed/ Static	0x00	The access network does not provide any suggestion to the access terminal about inter-cell handoff requirements.
			0x01–0xff	The access terminal should not perform two Inter-Cell FL or RL switches within InterCellCQISwitchTimer. This field is in units of 25 ms.
0x0002	MaxCDMASubsegmentsPerInterface Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x02	The maximum number of CDMA subsegments that the access terminal can transmit on in a given PHY frame.
			0x03–0x20	The maximum number of CDMA subsegments that the access terminal can transmit on in a given PHY frame.
			All other values	Reserved
0x0003	MaxODCCHReportsPerInterface Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x01	The maximum number of ODCCH reports that the access terminal can transmit on in a given PHY frame.
			0x02–0x20	The maximum number of ODCCH reports that the access terminal can transmit on in a given PHY frame.
			All other values	Reserved.
0x0004	MIMOResponseMessageThreshold	Soft Allowed/ Static	0x03	Value of this field in dB.
			0x00–0x02 and 0x04–0x10	Value of this field in dB.
			All other values	Reserved.
0x0005	MIMOResponseMessageTimer	Soft Allowed/ Static	0x05	Value of this field in seconds.
			0x01–0x04 and 0x06–0xff	Value of this field in seconds.
			0x00	The access terminal is not required to send MIMOResponse messages.
0x0006	REQBackoffResetTime	Soft Allowed/ Static	0x08	REQ backoff reset time in units of eight PHY frames.
			0x00–0x07 and 0x09–0xff	REQ backoff reset time in units of eight PHY frames.

Table 164—Simple configuration attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0007	REQBackoffFactor	Soft Allowed/ Static	0x40	REQBackoffFactor, in units of 0.01.
			0x00–0x09 and 0x65–0xff	Reserved
			All other values	REQBackoffFactor, in units of 0.01.
0x0008	CQIBackoffResetTime	Soft Allowed/ Static	0x08	CQI backoff reset time in units of eight PHY frames.
			0x00–0x07 and 0x09–0xff	CQI backoff reset time in units of eight PHY frames.
0x0009	CQIBackoffFactor	Soft Allowed/ Static	0x40	CQIBackoffFactor, in units of 0.01.
			0x00–0x09 and 0x65–0xff	Reserved.
			All other values	CQIBackoffFactor, in units of 0.01.
0x000a	NumNULLCQIValidity	Soft Allowed/ Static	0x30	Number of PHY frames after a flow control NULL CQI report during which the access network shall not schedule forward link data for the access terminal.
			0x08–0x2f and 0x31–0xff	Number of PHY frames after a flow control NULL CQI report during which the access network shall not schedule forward link data for the access terminal.
			All other values	Reserved.
0x000b	FLHandoffSupervisionTimeout	Soft Allowed/ Static	0x04	FL handoff supervision timeout in units of 100 ms.
			0x01–0x03 and 0x05–0x08	FL handoff supervision timeout in units of 100 ms.
			All other values	Reserved
0x000c	RLHandoffSupervisionTimeout	Soft Allowed/ Static	0x04	RL handoff supervision timeout in units of 100 ms.
			0x01–0x03 and 0x05–0x08	RL handoff supervision timeout in units of 100 ms.
			All other values	Reserved.

Table 164—Simple configuration attributes (continued)

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x000d	MaxNumSCCHacks Capability: access network shall not propose this attribute	Soft Allowed/Static	0x02	The number of SCCH blocks that the access terminal can acknowledge to one sector.
			0x03–0x08	The number of SCCH blocks that the access terminal can acknowledge to one sector.
			All other values	Reserved
0x01NN	StreamNNReqChannelCfg1QoSClass NN is the hexadecimal Stream ID extended to 8 bits by padding '0's on MSBs, where hexadecimal digits A through F are specified in uppercase letters.	Soft Allowed/Dynamic	0x00 For all other Streams	RequestQoS value to use for data belonging to Streams other than Stream 0–2.
			0x07 For streams 0–2	Default QoS class for signaling streams
			0x00–0x07	RequestQoS value to use for data belonging to Stream NN.
			All other values	Reserved.
0x02NN	StreamNNReqChannelCfg2QoSClass NN is the hexadecimal Stream ID extended to 8 bits by padding '0's on MSBs, where hexadecimal digits A through F are specified in uppercase letters.	Soft Allowed/Dynamic	0x00	RequestQoS value to use for data belonging to Streams other than Stream 0–2
			0x03 For streams 0–2	Default QoS class for signaling streams.
			0x00–0x03	RequestQoS value to use for data belonging to Stream NN.
			All other values	Reserved
0x03NN	StreamNNRequestPowerBoostAllowed NN is the hexadecimal Stream ID extended to 8 bits by padding '0's on MSBs, where hexadecimal digits A through F are specified in uppercase letters.	Soft Allowed/Dynamic	0x00	Request generated for packets in this Stream shall not be power boosted.
			0x01	Request generated for packets in this Stream may be power boosted.
			All other values	Reserved

8.7.7.2 Complex attributes

The complex attributes and default values are defined in Table 165 and the text that follows (see 15.2.4 for attribute record definition).

8.7.7.2.1 PowerControl attribute

This attribute has AttributeID 0x8000 (see Table 165).

Table 165—PowerControl attribute

Field	Length (bits)	Default Value
MaxPQICHErasures	4	0100
DefaultPQIValueForRLSS	4	1010
MinPQIValue	4	0010
MaxChanDiffImbalance	4	0110
MaxRLPilotDifferenceForFLSS	4	0110
MaxRLPilotDifferenceForDFLSS	4	0100
MaxRLPilotDifferenceForRLSS	3	011
MaxRLPilotDifferenceForDRLSS	3	001
OpenLoopTransitionTime	4	0000
MaxIOTCHErasures	4	0011
MaxInterferenceOffset	4	1111
PowerControlStepSizeAttribute	3	011
CQITargetCtoIAttribute	3	011
NullCQICtoI	2	011
HandoffCQICtoI	2	10
NSSCQIPilotOffset	3	100
REQTargetCtoIAttribute	3	101
REQQoSPowerBoost	2	10
HandoffREQCtoI	2	10
PAHTargetCtoIAttribute	3	101
PSDTargetCtoIAttribute	3	101
ODCTargetCtoIAttribute	3	011
ACKTargetCtoIAttribute	3	100
ACKStepUpSize	8	100
MinPAHChangeAttribute	3	010
MinChanDiffChangeAttribute	3	010

MaxPQICHErasures This field specifies the number of F-PQICH erasures during which the access terminal holds the most recent unerased value. A zero value for this field shall be mapped to infinity.

DefaultPQIValueForRLSS This field specifies a default PQI value for RLSS to be used when the number of consecutive erasures of F-PQICH from RLSS is larger than MaxPQICHErasures. The value of this field shall be interpreted using the mapping described in Table 138.

MinPQIValue	This field specifies the minimum value below which the reported PQI values will be deemed unreliable. The value of this field shall be interpreted using the mapping described in Table 138.
MaxChanDiffImbalance	This field specifies a maximum difference between FL and RL ChanDiff values, above which the access terminal shall use FL ChanDiff to compute pCoT, and shall be in units of dB.
MaxRLPilotDifferenceForDFLSS	This parameter specifies the maximum allowable difference in the RL pilot qualities of the best RL sector and candidate DFLSS, and shall be in units of dB.
MaxRLPilotDifferenceForRLSS	This parameter specifies the maximum allowable difference in the RL pilot qualities of the best RL sector and the current RLSS, and shall be in units of dB.
MaxRLPilotDifferenceForDRLSS	This parameter specifies the maximum allowable difference in the RL pilot qualities of the best RL sector and candidate DRLSS, and shall be in units of dB.
OpenLoopTransitionTime	This parameter specifies the transition time for open-loop power adjustment. A zero value indicates that the open-loop power adjustment procedures shall not be performed. The value of this parameter shall be set to $8n$ PHY frames, where n is the value of the four-bit field.
MaxIOTCHerasures	This parameter specifies the maximum number of F-IOTCH erasures after which the access terminal will use SlowInterferenceOffset reports in the ExtendedChannelInfo.
MaxInterferenceOffset	This parameter defines the maximum for the InterferenceOffset values that access terminal uses for adjusting the transmit power density of several RL channels, and shall be in units of dB.
PowerControlStepSizeAttribute	This parameter specifies the step size for closed-loop power control of R-PICH, and shall take the value $0.25 + 0.25n$ dB, where n is the value of the three-bit field.
CQITargetCtoIAttribute	This parameter specifies a C/I target for r-cqich transmissions on R-CDCCH, and shall take the value $-22 + n$ dB, where n is the value of the three-bit field.
NullCQICtoI	This parameter specifies a C/I target for NULL r-cqich transmissions on R-CDCCH, and shall take the value $CQITargetCtoI + \Delta$ dB, where Δ takes values 0, 1, 3, or 5 dB, when the two-bit field is set to '00', '01', '10', or '11', respectively.
HandoffCQICtoI	This parameter specifies a C/I target for handoff r-cqich transmissions on R-CDCCH, and shall take the value $CQITargetCtoI + \Delta$ dB, where Δ is 0, 1, 3, or 5 dB when the two-bit field is set to '00', '01', '10', or '11' respectively.
NSSCQIPilotOffset	This parameter specifies the power offset of r-cqich transmissions on R_CDCCH to non-serving sectors with respect to R-PICH, and shall take the values 0, 1, 2, 3, 5, 7, 9, or 11 dB when the three-bit field is set to '000', '001', '010', '011', '100', '101', '110', or '111', respectively.

REQTargetCtoIAttribute	This parameter specifies a C/I target for r-reqch transmissions on R-CDCCH, and shall take the value $-21 + n$ dB, where n is the value of the three bit field.
REQQoSPowerBoost	This parameter specifies a power boost value for r-reqch transmissions on R-CDCCH for flows that allow request boost, and shall take the values 0, 1, 3, or 5 dB when the two-bit field is set to '00', '01', '10', or '11', respectively.
HandoffREQCtoI	This parameter specifies a C/I target for handoff r-reqch transmissions on R-CDCCH, and shall take the value REQTargetCtoI + Delta dB, where Delta is 0, 1, 3, or 5 dB when the two-bit field is set to '00', '01', '10', or '11', respectively.
PAHTargetCtoIAttribute	This parameter specifies a C/I target for r-pahch transmissions on R-CDCCH, and shall take the value $-21 + n$ dB, where n is the value of the three-bit field.
PSDTargetCtoIAttribute	This parameter specifies a C/I target for r-psdch transmissions on R-CDCCH, and shall take the value $-22 + n$ dB, where n is the value of the three-bit field.
ODCTargetCtoIAttribute	This parameter specifies a C/I target for R-ODCCH transmissions, and shall take the value $-3 + n$ dB, where n is the value of the three-bit field.
ACKTargetCtoIAttribute	This parameter specifies a C/I target for R-ACKCH transmissions, and shall take the value $-4 + n$ dB, where n is the value of the three-bit field.
ACKStepUpSize	This parameter specifies the step size for adjusting the power density of the R-ACKCH, and shall take the value $n \times 0.25$ dB, where n is the value of the three-bit field
MinPAHChangeAttribute	This parameter specifies the minimum change in the PA headroom value after which the access terminal is allowed to transmit an r-pahch report on R-CDCCH, and shall be in units of dB.
MinChanDiffChangeAttribute	This parameter specifies the minimum change in the ChanDiff value after which the access terminal is allowed to transmit an r-psdch report on R-CDCCH, and shall be in units of dB.

8.7.8 Protocol numeric constants

The protocol numeric constants are listed in Table 166.

Table 166—Numeric constants of the Reverse Control MAC Protocol

Constant	Meaning	Value
$N_{RCCMPTType}$	Type field for this protocol	See 16.1
$N_{RCCMPBasic}$	Subtype field for this protocol	0x00

8.7.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

8.8 Basic Reverse Traffic Channel MAC Protocol

8.8.1 Overview

The Basic Reverse Traffic Channel MAC Protocol provides the procedures and messages required for an access terminal to transmit, and for an access network to receive, the Reverse Traffic Channel. The access network maintains an instance of this protocol for every access terminal.

This protocol operates in one of two states:

- *Inactive State*: In this state, the access terminal is not assigned a MACID and cannot transmit on the Reverse Traffic Channel. When the protocol is in this state, it waits for an *Activate* command.
- *Active State*: In this state, the access terminal is assigned a MACID and may transmit data on the Reverse Traffic Channel.

The protocol states and the indications and events causing the transition between the states are shown in Figure 61.

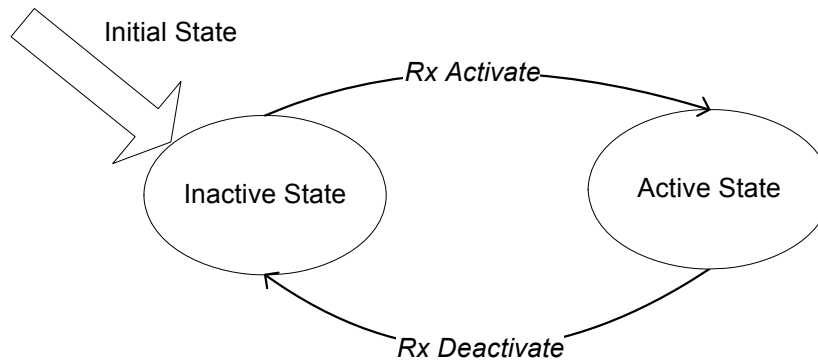


Figure 61—State Diagram for Basic Reverse Traffic Channel MAC Protocol

This protocol uses the following parameters and attributes listed in Table 167, where OMP and ASMP refer to the Overhead Messages Protocol and Active Set Management Protocol, respectively.

Table 167—Parameters and Attributes for the Reverse Traffic Channel MAC Protocol

Parameter Name	Where defined	Comments
AttemptBoost _j , $j=1,2,\dots,6$	Configuration attribute	—
AuxPICHOffset	Configuration attribute	—
CDCHSubsegment _j , $j = 1,\dots,\text{NumCDCHSubsegments}$	ASMP	Public data
CDMATrafficPilotOffset _p , $p = 0,1$	Configuration attribute	—
ChanDiffMax	Configuration attribute	—

**Table 167—Parameters and Attributes for the
Reverse Traffic Channel MAC Protocol (continued)**

Parameter Name	Where defined	Comments
ChanDiffMin	Configuration attribute	—
DataCoTMax	Configuration attribute	—
DataCoTMin	Configuration attribute	—
DownDecisionThresholdMin	Configuration attribute	—
DownDecisionValue	Configuration attribute	—
DwDownThreshold	Configuration attribute	—
DwUpThreshold	Configuration attribute	—
DuplexingMode	OMP	Public data
EffectiveTransmitPower	OMP	ExtendedChannelInfo
EraseTargetCtoI _j , j=0,1,2,3	OMP	ExtendedChannelInfo
FastOSIEnabled	OMP	ExtendedChannelInfo
FastOSIMonitorSetSize	Configuration attribute	—
FastOSIMonitorThreshold	Configuration attribute	—
FastOSIStepDown	Configuration attribute	—
FastOSIStepUp	Configuration attribute	—
InBandPAHTimeout	Configuration attribute	—
InBandREQTimeout	Configuration attribute	—
LDPCSupportedRL	ASMP	Public data
MinDataCtoIChange	Configuration attribute	—
MinDeltaSlowChange	Configuration attribute	—
NumCDCHSubsegments	ASMP	Public data
NCRCData	PHY	Public data
OSI2SequenceNumMax	Configuration attribute	—

**Table 167—Parameters and Attributes for the
Reverse Traffic Channel MAC Protocol (continued)**

Parameter Name	Where defined	Comments
OSIMonitorSetSize	Configuration attribute	—
OSIMonitorThreshold	Configuration attribute	—
OSIResponseMode	OMP	ExtendedChannelInfo
OSIStepDown	Configuration attribute	—
OSIStepUp	Configuration attribute	—
PacketStartInterlace _i , $i = 0,1$	ASMP	Public data CDCHAssignmentMsg
RABEnabled	OMP	ExtendedChannelInfo
RABMonitorThreshold	Configuration attribute	—
RLNumSDMADimensions	OMP	ExtendedChannelInfo
TotalNumSubCarriers	OMP	SystemInfo
TuneAwayStatus	Connected State Protocol	Public data
UpDecisionThresholdMin	Configuration attribute	—
UpDecisionValue	Configuration attribute	—

8.8.2 Primitives

8.8.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

8.8.2.2 Return indications

This protocol returns the following indications:

- *ReverseTrafficPacketsMissed*
- *SupervisionFailed*

8.8.2.3 Local common data

- InBandRequestTimer Active
- Subtype for this protocol

8.8.2.4 Public data

This protocol does not define any static public data.

8.8.3 Protocol data unit

The transmission unit of this protocol is a Reverse Traffic Channel MAC layer packet. Each packet contains one consolidated packet.

8.8.4 Procedures and messages for the InConfiguration instance of the protocol

8.8.4.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

8.8.4.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

8.8.4.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

8.8.5 Procedures for the InUse instance

This protocol constructs a Reverse Traffic Channel MAC packet out of the Consolidated Packet received from the Packet Consolidation Protocol.

The protocol then sends the packet for transmission to the Physical Layer.

8.8.5.1 Initialization and commit procedures

8.8.5.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the InActive State.
- The protocol at the access network shall enter the InActive State.

8.8.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

8.8.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

8.8.5.2 Command processing

8.8.5.2.1 Activate

If the protocol receives an *Activate* command in the Inactive State, the access terminal and the access network shall perform the following:

- Transition to the Active State
- Internal variables of the protocol shall be initialized as mentioned in the definition of these variables.

If the protocol receives this command in any other state, the command shall be ignored.

8.8.5.2.2 Deactivate

If the protocol receives a *Deactivate* command in the Active State:

- The access terminal shall cease transmitting the Reverse Traffic Channel and shall transition to the Inactive State.
- The access network shall cease monitoring the Reverse Traffic Channel from this access terminal and shall transition to the Inactive State.

If the protocol receives a *Deactivate* command in the Inactive State, the command shall be ignored.

8.8.5.3 Reverse Traffic Channel addressing

Transmission on the Reverse Traffic Channel is multiplexed in time and frequency. An assignment on the Reverse Traffic Channel shall be specified by a set of hop-ports and a set of PHY frames. Each hop-port is specified by a hop-port index.

The duration of an assignment of hop-ports may or may not be pre-specified. Assignments whose durations are pre-specified are known as non-persistent assignments, and assignments whose durations are not pre-specified are known as persistent assignments.

The set of hop-ports assigned for a given set of PHY frames whose duration is not pre-specified is referred to as the “Reverse Link Access Terminal Assignment” or RL-ATA. RL-ATAs are assigned via RLABs that have the persistent bit set to '1', and may be altered for a specified length of time via a RLAB with the persistent bit set to '0'. (See 8.8.5.5.1.3.2.1 for more details.) An access terminal can have multiple RL-ATAs, as long as they do not overlap in time.

The set of hop-ports assigned for a given set of PHY frames for a particular access terminal whose duration is pre-specified is referred to as the “Reverse Link Non-persistent Access Terminal Assignment” or RL-NP-ATA. The RL-NP-ATA is assigned via RLABs that have the persistent bit set to '0', which are denoted as NP-RLABs. An access terminal can have multiple RL-NP-ATAs, as long as they do not overlap in time.

Sets of hop-ports in assignment blocks received from the FLCS MAC protocol, described in 8.8.5.9, are specified using the channel tree.

Packets transmitted over the Reverse Traffic Channel are transmitted over the R-ODCH Physical Layer channel. Access terminals are assigned R-ODCH resources (RL-ATAs, RL-NP-ATAs) via assignment blocks (RLABs, NP-RLABs) that are sent over the F-SCCH.

The following rules apply regarding the coexistence of RL-ATAs and RL-NP-ATAs:

- No two RL-ATAs/RL-NP-ATAs can overlap in time.

- An access terminal shall not have a non-empty RL-ATA that overlaps in time with a non-empty RL-NP-ATA.

Additionally, the access network shall ensure that the following rules hold:

- A RL-ATA/RL-NP-ATA shall not span multiple SDMA subtrees. Additionally, an RL-ATA/RL-NP-ATA shall not occupy a subtree with index larger than `MaxNumRLSDMASubtrees`.
- If `DuplexingMode = FDD`, the packet size of a RL-ATA/RL-NP-ATA shall not be greater than `max(MaxRLPacketSize, MaxRLPacketSizeLDPC)` or `MaxRLPacketSizeExtendedFrame`, depending on whether an 8-interlace structure without extended frames or an 8-interlace structure with extended frames is being used. Additionally, if the ASMP parameter `LDPCEnabledRL` is set to '0', the packet size of an RL-ATA/RL-NP-ATA shall not be greater than `MaxRLPacketSize` when an 8-interlace structure is being used. `DuplexingMode` is public data of the OMP.
- If `DuplexingMode = FDD`, the number of usable hop-ports in an RL-ATA/RL-NP-ATA shall not be greater than `MaxRLAssignmentSize` or `MaxRLAssignmentSizeExtendedFrame`, depending on whether an 8-interlace structure without extended frames or an 8-interlace structure with extended frames is being used.
- If `DuplexingMode = TDD44`, the packet size of a RL-ATA/RL-NP-ATA shall not be greater than `max(MaxRLPacketSizeTDD44, MaxRLPacketSizeLDPCTDD44)` or `MaxRLPacketSizeExtendedFrameTDD44`, depending on whether an interlacing structure without or with extended frames is being used. Additionally, if the ASMP parameter `LDPCSupportedRL` is set to '0', the packet size of an RL-ATA/RL-NP-ATA shall not be greater than `MaxRLPacketSizeTDD44`.
- If `DuplexingMode = TDD44`, the number of usable hop-ports in an RL-ATA/RL-NP-ATA shall not be greater than `MaxRLAssignmentSizeTDD44` or `MaxRLAssignmentSizeExtendedFrameTDD44`, depending on whether an interlace structure without or with extended frames is being used.
- If `DuplexingMode = TDD63`, the packet size of a RL-ATA/RL-NP-ATA shall not be greater than `max(MaxRLPacketSizeTDD63, MaxRLPacketSizeLDPCTDD63)` or `MaxRLPacketSizeExtendedFrameTDD63`, depending on whether an interlacing structure without or with extended frames is being used. Additionally, if the ASMP parameter `LDPCSupportedRL` is set to '0', the packet size of an RL-ATA/RL-NP-ATA shall not be greater than `MaxRLPacketSizeTDD63`.
- If `DuplexingMode = TDD63`, the number of usable hop-ports in an RL-ATA/RL-NP-ATA shall not be greater than `MaxRLAssignmentSizeTDD63` or `MaxRLAssignmentSizeExtendedFrameTDD63`, depending on whether an interlace structure without or with extended frames is being used.

All hop-ports in the RL-ATA/RL-NP-ATA for an access terminal for a given PHY frame shall be combined for transmission over the Physical Layer channel (R-ODCH).

8.8.5.4 Inactive State

When the protocol is in the Inactive State, the access terminal and the access network wait for an *Activate* command.

8.8.5.5 Active State

In this subclause, the parameters `RDCHGainMin` and `RDCHGainMax` correspond to the RLSS.

In the Active State, the access terminal shall transmit over the R-ODCH using the RL-ATAs or RL-NP-ATAs and PFs selected by the access network and signaled to the access terminal over the F-SCCH Physical Layer channel. The access terminal processes blocks from the FLCS MAC Protocol to maintain its RL ATA and RL-NP-ATA, and configures the Physical Layer for transmission of packets according to the RL-ATA/RL-NP-ATA. The access network controls transmission of ACK/NACK information via the FLCS MAC

Protocol (over the F-SCCH Physical Layer channel) based on pass or fail of the Lower MAC packet, as determined by the PHY.

Note that an RL-NP-ATA shall be used for transmission of a single Lower MAC packet.

The MACID assigned to the access terminal for each sector in its active set is given in the MACResourceAssignment message that is public data of the Active Set Management Protocol.

OSI2SequenceNum and PilotPNStrongest are an access terminal's locally maintained parameters. The initial value of OSI2SequenceNum shall be equal to 1, and the initial value of PilotPNStrongest shall be equal to -1.

RDCHGain is an access terminal's locally maintained parameter, whose initial value shall be equal to RDCHGainMin.

Assignment and HARQ logic for the access network and access terminal are specified in 8.8.5.5.1, 8.8.5.5.2, and 8.8.5.5.3.

8.8.5.5.1 Access terminal requirements

If any RLAB/NP-RLAB with NodeID NodeID_{DEASSIGN}, as defined in 8.8.5.9, is received by an access terminal when it is in semi-connected state, this shall initiate a transition to the connected state.

8.8.5.5.1.1 Access terminal rules for processing assignment blocks

This subclause specifies the rules for interpreting and combining the reverse link assignment blocks that are received in each FL PHY frame from the FLCS MAC protocol. These assignment blocks are scrambled with the MACID of the access terminal.

In the following, "x-RLAB" refers to any of the following: RLAB or NP-RLAB. Similarly, "x-ATA" refers to any of the following: RL-ATA or RL-NP-ATA. The following procedure for ordering and processing x-RLABs shall be applied separately to the set of RLABs and the set of NP-RLABs:

- a) If any x-RLAB received from the FLCS MAC protocol has the NodeID field set to NodeID_{DEASSIGN}, has the same type as the corresponding existing x-ATA (i.e., RL-ATA if the received x-RLAB is an RLAB, RL-NP-ATA if the received x-RLAB is an NP-RLAB_{UC}), and has the PF field set to the current packet format of the corresponding existing x-ATA, then this x-RLAB shall be treated as an x-RLAB for deassignment of the corresponding x-ATA. Deassignment is described for RL-ATAs and RL-NP-ATAs in 8.8.5.5.1.3.1 and 8.8.5.5.1.4.1 respectively. In addition, any other simultaneously received x-RLABs shall be discarded. The subsequently described steps in this subclause shall not be carried out in this case. Additionally, if an F-ACKCH value of 2 or 3 is received from the FLCS MAC Protocol in this PHY frame, this shall be treated as a deassignment of the x-ATA. Any other simultaneously received RLABs or NP-RLABs shall be discarded.
- b) If multiple x-RLABs are received from the FLCS MAC protocol in the same FL PHY frame, these x-RLABs are ordered according to the F-SCCH index pair (i, j) defined in 8.5.5.4.1.1.3.6. An x-RLAB (i, j) is said to have a lower index pair than an x-RLAB (i', j') if either $i < i'$, or $i = i'$ and $j < j'$. The set of hop-ports of the x-RLAB with the lowest index pair is denoted by N_0 , the set of hop-ports of the x-RLAB with the next higher index pair is denoted by N_1 , and so on. The set of hop-ports indexed by an x-RLAB is given either by the NodeID field of the x-RLAB. In the above, a NodeID equal to NodeID_{DEASSIGN} is mapped to the null set of hop-ports, denoted by Φ .
- c) If multiple x-RLABs are received from the FLCS MAC protocol in the same FL PHY frame and these x-RLABs do not have the same values in all fields except the NodeID and Supplemental fields, the access terminal shall determine that an error has occurred. In this case, the access terminal shall

ignore all these x-RLABs. Additionally, the remaining steps shown in this subclause shall not be carried out.

- d) If all the received x-RLABs have a common packet format field of NULL, the access terminal shall carry out the following operations:
 - 1) If all the received x-RLABs do not have a common value for the Supplemental field, then the access terminal shall determine that an error has occurred. In this case, the access terminal shall ignore all these x-RLABs.
 - 2) If all the received x-RLABs have the Supplemental field set to '0', the access terminal shall treat this set of x-RLABs as a single x-RLAB for Resource Reassignment. The final set of hop-ports after the Resource Reassignment operation shall be given by $f_{\text{ATA\_Computation}}(\Phi, N_0, N_1, \dots)$, where the function $f_{\text{ATA\_Computation}}$ is as defined in 8.8.5.5.1.1.1 and Φ denotes the empty set of hop-ports. In the above, a set of RLABs causes Resource Reassignment of an RL-ATA and a set of NP-RLABs causes Resource Reassignment of an RL-NP-ATA. The Resource Reassignment operation for RL-ATAs and RL-NP-ATAs is described in 8.8.5.5.1.3.3 and 8.8.5.5.1.4.3, respectively.
 - 3) If all the received x-RLABs have the Supplemental field set to '1', the access terminal shall treat this set of x-RLABs as a single x-RLAB for HARQ extension. The final set of hop-ports after the HARQ extension operation shall be given by N_{init} if $N_0 = \Phi$, and by $f_{\text{ATA\_Computation}}(\Phi, N_0, N_1, \dots)$ otherwise. Here, the function $f_{\text{ATA\_Computation}}$ is as defined in 8.8.5.5.1.1.1, Φ denotes the empty set of hop-ports, and N_{init} denotes the set of hop-ports in the current x-ATA. In the above, a set of RLABs causes HARQ extension of an RL-ATA and a set of NP-RLABs causes HARQ extension of an RL-NP-ATA. The HARQ extension operation for RL-ATAs and RL-NP-ATAs is described in 8.8.5.5.1.3.4 and 8.8.5.5.1.4.4, respectively.
- e) If all the received x-RLABs have a common packet format field that is not equal to NULL, the access terminal shall carry out the following operations:
 - 1) If any x-RLAB except the one with the lowest index pair has the Supplemental field set to '0', the access terminal shall determine that an error has occurred. In this case, the access terminal shall ignore all these x-RLABs.
 - 2) If the x-RLAB with the lowest index pair has its supplemental field set to '0' and all the remaining x-RLABs have their supplemental field set to '1', the access terminal shall treat this as a new assignment (non-supplemental assignment in the case of RL-ATAs). The set of hop-ports assigned by this non-supplemental assignment shall be given by $f_{\text{ATA\_Computation}}(\Phi, N_0, N_1, \dots)$, where the function $f_{\text{ATA\_Computation}}$ is as defined in 8.8.5.5.1.1.1 and Φ denotes the empty set of hop-ports.
 - 3) If all the received x-RLABs have their supplemental field set to '1', the access terminal shall treat this as a supplemental assignment to an RL-ATA. Supplemental assignments modify an existing RL-ATA and are described in 8.8.5.5.1.3.2.1. The final set of hop-ports after the application of this supplemental assignment shall be given by $f_{\text{ATA\_Computation}}(N_{\text{init}}, N_0, N_1, \dots)$ where N_{init} denotes the set of hop-ports in the existing RL-ATA and the function $f_{\text{ATA\_Computation}}$ is as defined in 8.8.5.5.1.1.1.

8.8.5.5.1.1.1 Definition of $f_{\text{ATA\_Computation}}$

The function $f_{\text{ATA\_Computation}}$ takes a set of at least two input arguments in order, each of which denotes a set of hop-ports, and provides an output that is another set of hop-ports. The input set of hop-ports may be represented by a NodeID field of an F-SCCH block.

The output of the function $f_{\text{ATA\_Computation}}$ is computed according to the following procedure:

- a) Let n denote the number of input arguments to the function $f_{\text{ATA\_Computation}}$. Label the different arguments $T_0, T_1, \dots, T_{n-1}$. Initialize a counter i to 1, and a hop-port set T_{out} to T_0 .

- b) If T_i is a subset of T_{out} , then set T_{out} to be the set of hop-ports that are contained in T_{out} but not in T_i ; else set T_{out} to be the union of T_i and T_{out} .
- c) Increment i . If $i < n$, go to step b).

The output of the function $f_{ATA\_Computation}$ is given by the hop-port set T_{out} at the end of the above procedure.

8.8.5.5.1.2 Access terminal rules for overlapping assignments

The following rules shall be followed in order to satisfy the constraints specified in 8.8.5.3. Here, a new ATA refers to an ATA that has been created in this FL PHY frame and excludes modifications to an existing ATA through supplemental assignments, Resource Reassignment, and HARQ extension operations.

- a) If a new RL-ATA overlaps in time with a previously existing RL-NP-ATA or vice-versa, the new ATA takes precedence and the already existing ATA is expired.
- b) If an RL-ATA is created in the same FL PHY frame as an RL-NP-ATA, then both ATAs shall be expired.

8.8.5.5.1.3 Access terminal assignment management for persistent assignments

Upon receiving a *TunedAway* indication from the Air Link Management Protocol, the access terminal shall expire all its RL-ATAs.

In this subclause, it is assumed that all the RLABs/NP-RLABs are being sent from the same serving sector, the RLSS. The logic for access terminal assignment management during handoff is found in 8.8.5.5.1.5.

The access terminal shall perform the following steps in each FL-PHY frame to update its RL-ATAs:

- a) Follow the procedure described in 8.8.5.5.1.1 for updating the RL-ATA and RL-NP-ATA based on the received assignment blocks.
- b) After the RL-ATA and the RL-NP-ATA have all been updated, follow the procedures described in 8.8.5.5.1.2 for handling overlapping assignments.
- c) Follow the procedure specified in 8.8.5.5.1.3.4 for acknowledging assignments.

8.8.5.5.1.3.1 Deassigning assignments

If a received RLAB or F-ACKCH indication causes deassignment of the RL-ATA, as determined in 8.8.5.5.1.1, then the RL-ATA for the given set of PHY frames shall be expired.

8.8.5.5.1.3.2 Supplemental and non-supplemental assignments

8.8.5.5.1.3.2.1 Supplemental assignments

An existing RL-ATA can have its set of hop-ports modified via supplemental assignments, as described in 8.8.5.5.1.1. The supplemental assignment can be persistent or non-persistent, depending on whether it is caused by RLABs or NP-RLABs, respectively. Supplemental assignments can also be used to change other fields of an existing RL-ATA such as packet format, without changing the resources.

If a persistent supplemental assignment is received for a particular set of PHY frames, then the following are true:

- a) If the RL-ATA for the given PHY frames is empty, or has a different SubtreeID than the received RLABs, the access terminal shall determine that an error condition has occurred and shall ignore the RLAB.

- b) If the RL-ATA for the given set of PHY frames is not empty, and has the same subtreeID as the received RLABs, then the following are true:
 - 1) The hop-ports of the RL-ATA shall be modified as described in 8.8.5.5.1.1.
 - 2) The PF specified in the received RLABs shall be used in place of any PFs that may have been specified in any previous assignments. Similarly, the new subtreeID, RL Pilot, and RLPC will be used.

If a non-persistent supplemental assignment is received for a particular set of PHY frames, then the following are true:

- a) If the RL-ATA for the given PHY frames is empty, or has a different SubtreeID than the received NP-RLABs, the access terminal shall determine that an error condition has occurred and shall ignore the NP-RLAB.
- b) If the RL-ATA for the given PHY frames is not empty, and has the same subtreeID as the received NP-RLABs, then the following changes shall be made for the duration of only one packet, and the original RL-ATA shall take effect after an ACK is received.
 - 1) The hop-ports of the RL-ATA shall be modified as described in 8.8.5.5.1.1.
 - 2) The PF specified in the received NP-RLAB shall be used in place of any PFs that may have been specified in any previous assignment of NodeIDs (hop-ports). Similarly, the new subtreeID, RL Pilot, and RLPC will be used.

8.8.5.5.1.3.2.2 Non-supplemental assignments

If a non-supplemental assignment is received, as determined in 8.8.5.5.1.1, then the RL-ATA for set of PHY frames assigned by the received RLABs shall be cleared before adding the hop-ports specified by the set of received RLABs, as determined in 8.8.5.5.1.1 to the RL-ATA for the given set of PHY frames.

The PF, SubtreeID, RLPilot, and RLPC fields in the received RLABs shall be used for the new RL-ATA.

8.8.5.5.1.3.3 Resource Reassignment

If a Resource Reassignment is received for a particular set of PHY frames, as described in 8.8.5.5.1.1, then the following are true:

- a) If the RL-ATA for the given set of PHY frames is empty, or if the received RLABs have a different subtreeID or RL Pilot than previously used for the RL-ATA, the access terminal shall determine that an error condition has occurred and shall ignore the RLAB.
- b) If the RL-ATA for the given set of PHY frames is not empty, and the received RLABs have the same subtreeID and RL Pilot as previously used for the RL-ATA, then the RL-ATA shall be changed to the set of hop-ports determined in 8.8.5.5.1.1 for the subsequent HARQ retransmissions. The same PF will be used as previously used for the RL-ATA.

8.8.5.5.1.3.4 HARQ extension

A persistent assignment can have the total number of HARQ transmissions increased for one packet via HARQ extension.

If a HARQ extension is received for a particular set of PHY frames as determined in 8.8.5.5.1.1, then the following are true:

- a) If the RL-ATA for the given set of PHY frames is empty, or if the received RLABs have a different subtreeID or RL Pilot than previously used for the RL-ATA, the access terminal shall determine that an error condition has occurred and shall ignore the received RLABs.

- b) If the RL-ATA for the given set of PHY frames is not empty, and the received RLABs have the same subtreeID and RL Pilot as previously used for the RL-ATA, then there will be six additional HARQ transmissions for the packet currently being transmitted on the RL-ATA, starting with the first HARQ transmission after the HARQ extension was received. Additionally, the set of hop-ports shall be moved to the set of hop-ports determined in 8.8.5.5.1.1. The same PF will be used as previously used for the RL-ATA.

HARQ transmissions of a given packet can be extended multiple times, using HARQ extension assignment blocks.

8.8.5.5.1.3.5 Assignment acknowledgment

The access terminal shall issue a *ReverseControlChannelMAC.ACKAssignment* along with the index k if the RLAB index pair (as defined in 8.5.5.4.1.1.3.6) is equal to $(k, 0)$ and the following conditions are satisfied (note that multiple such commands may be issued in each FL PHY frame):

- a) The access terminal received an RLAB with index pair $(k, 0)$, where the index pair is as defined in 8.5.5.4.1.1.3.6.
- b) The value of AssignmentAckHARQtransmission is not 0.
- c) There have been AssignmentAckHARQtransmission HARQ transmissions on the changed RL-ATA.
- d) The R-DCH packet being transmitted on the RL-ATA has not been successfully decoded.

Assignment acknowledgments follow the same timing structure as ACKs.

- If DuplexingMode = FDD for FL 8-interlace structure without extended frames (see 8.8.5.5.1.1), i.e., if the RLAB is received on FL PHY frame k , the acknowledgment can be transmitted on RL PHY frames $k + 5 + 8n$ where $n = 0, 1, \dots, 5$, subject to the other conditions specified in this subclause.
- For DuplexingMode = TDD44 and 4-interlace structure with or without extended frames, if the RLAB is received on FL PHY frame k , the acknowledgment can be transmitted on RL PHY frames $k + 4n$ where $n = 0, \dots, 5$, subject to the other conditions specified in this subclause.
- For DuplexingMode = TDD63 and 3-RL interlace structure with or without extended transmissions, if the RLAB is received on FL PHY frame $2k + 1$, the acknowledgment can be transmitted on RL PHY frames $k + 3n$ where $n = 0, \dots, 5$, subject to the other conditions specified in this subclause.

8.8.5.5.1.4 Access terminal assignment management for non-persistent assignments

Upon receiving a *TunedAway* indication from the Connected State Protocol, the access terminal shall expire all its RL-NP-ATAs.

In this subclause, it is assumed that all the RLABs/NP-RLABs are being sent from the same serving sector, namely the RLSS. The logic for access terminal assignment management during handoff is found in 8.8.5.5.1.5.

The access terminal shall give up a RL-NP-ATA after it has kept it for the maximum number of HARQ transmissions, unless it receives assignment blocks for HARQ extension, as determined in 8.8.5.5.1.1.

The access terminal shall perform the following steps in each FL PHY frame to update its RL-NP-ATAs:

- a) Follow the procedure described in 8.8.5.5.1.1 for updating the RL-ATA and the RL-NP-ATA.
- b) After the RL-ATA and the RL-NP-ATA have been updated, follow the procedures described in 8.8.5.5.1.2 for handling overlapping assignments.
- c) Follow the procedure specified in 8.8.5.5.1.4.4 for acknowledging assignments.

8.8.5.5.1.4.1 Deassigning assignments

If a received RLAB or F-ACKCH indication causes deassignment of the RL-NP-ATA, as determined in 8.8.5.5.1.1, then the RL-NP-ATA for the given set of PHY frames shall be expired.

8.8.5.5.1.4.2 New non-persistent assignments

If a new non-persistent assignment is received, as determined in 8.8.5.5.1.1, then the RL-NP-ATA for set of PHY frames assigned by the received RLABs shall be cleared before adding the hop-ports specified by the set of received RLABs, as determined in 8.8.5.5.1.1 to the RL-NP-ATA for the given set of PHY frames.

The PF, SubtreeID, RLPilot, and RLPC fields in the received RLABs shall be used for the new RL-NP-ATA.

8.8.5.5.1.4.3 Resource Reassignment

If a Resource Reassignment is received, as determined in 8.8.5.5.1.1, for a particular set of PHY frames, then the following are true:

- a) If the RL-NP-ATA on the given set of PHY frames is empty, or if the received NP-RLABs have a different subtreeID or RL Pilot than previously used for the RL-NP-ATA, the access terminal shall determine that an error condition has occurred and shall ignore the received NP-RLABs.
- b) If the RL-NP-ATA on the given set of PHY frames is not empty, and the received NP-RLABs have the same subtreeID and RL Pilot as previously used for the RL-NP-ATA, then the RL-NP-ATA shall be changed to the set of hop-ports determined in 8.8.5.5.1.1 for the subsequent HARQ transmissions. The same PF will be used as previously used for the RL-NP-ATA.

8.8.5.5.1.4.4 HARQ extension

A non persistent assignment can have the total number of HARQ transmissions increased via HARQ extension.

If a HARQ extension is received for a particular set of PHY frames, as determined in 8.8.5.5.1.1, then the following are true:

- a) If the RL-NP-ATA for the given PHY frames is empty, or if the received NP-RLABs have a different subtreeID or RL Pilot than previously used for the RL-NP-ATA, the access terminal shall determine that an error condition has occurred and shall ignore the received NP-RLABs.
- b) If the RL-NP-ATA for the given PHY frames is not empty, and the received NP-RLABs have the same subtreeID and RL Pilot as previously used for the RL-NP-ATA, then there will be six additional HARQ transmissions for the packet currently being sent on the RL-NP-ATA, starting with the first HARQ transmission after the HARQ extension was received. Additionally, the set of hop-ports shall be moved to the set of hop-ports determined in 8.8.5.5.1.1. The same PF will be used as previously used for the RL-NP-ATA.

HARQ transmissions of a given packet can be extended multiple times, using HARQ extension assignment blocks.

8.8.5.5.1.4.5 Assignment acknowledgement

The access terminal shall issue a *ReverseControlChannelMAC.ACKAssignment* along with the index k if the following conditions are satisfied (note that multiple such commands may be issued in each FL PHY frame):

- a) The access terminal received an FLAB with index pair (k, 0), where the index pair is as defined in 8.5.5.4.1.1.3.6.

- b) The value of AssignmentAckHARQtransmission is not 0.
- c) There have been AssignmentAckHARQtransmission HARQ transmissions on the changed RL-NP-ATA.
- d) The R-DCH packet being transmitted on the RL-NP-ATA has not been successfully decoded.

Assignment acknowledgments follow the same timing structure as ACKs.

- If DuplexingMode = FDD for FL 8-interlace structure without extended frames (see 8.1.5.1.1), i.e., if the RLAB is received on FL PHY frame k , the acknowledgment can be transmitted on RL PHY frames $k + 5 + 8n$ where $n = 0, 1, \dots, 5$, subject to the other conditions specified in this subclause.
- For DuplexingMode = TDD44 4-interlace structure, if the RLAB is received on FL PHY frame k , the acknowledgment can be transmitted on RL PHY frames $k + 4n$ where $n = 0, \dots, 5$, subject to the other conditions specified in this subclause.
- For DuplexingMode = TDD63 and 3-RL interlace structure with or without extended transmissions, if the RLAB is received on FL PHY frame $2k + 1$, the acknowledgment can be transmitted on RL PHY frames $k + 3n$ where $n = 0, \dots, 5$, subject to the other conditions specified in this subclause.

8.8.5.5.1.5 Access terminal assignment management during handoff

If an *RLSSChanged* indication from the RCC MAC Protocol is received, the access terminal shall clear all RL-ATAs and RL-NP-ATAs associated with the old RLSS.

If the access terminal receives an RLAB/NP-RLAB, that has the supplemental field set to '0', from the DRLSS, while the DRLSS is different from the RLSS, the access terminal shall issue a *ReverseControlChannelMAC.ChangeRLSS* command to change from the RLSS to the DRLSS. If the access terminal then receives an *RLSSChanged* indication from the RCC MAC Protocol, the access terminal shall process the *RLSSChanged* indication as specified above, and in addition shall update the appropriate RL-ATA/RL-NP-ATA according to the new RLAB/NPRLAB.

The access terminal shall ignore all RLABs or NP-RLABs that come from sectors other than its current RLSS or its DRLSS.

8.8.5.5.1.6 Access terminal transmission logic for persistent assignments

The access terminal may formulate and transmit a Lower MAC packet on a PHY frame according to the RL-ATA on that PHY frame and the PF selected for transmission. The Lower MAC shall pass the Physical Layer Protocol the set of hop-ports specified by the RL-ATA. The power for transmission of a particular Lower MAC packet is computed as specified in 8.8.5.5.1.9. The access network shall inform the access terminal of its assignment using signaling messages as described in the FLCS MAC Protocol specification.

If the packet format for the RL-NP-ATA matches one of the configuration attributes *RLMultiDecodeAssignedPFK*, and if the size of the RL-NP-ATA in base nodes is equal to configuration attribute *RLMultiDecodeAssignmentSizeK*, then the access terminal may transmit one of multiple packet formats, as specified by configuration attributes *RLMultiDecodeCompatiblePFKj* where j goes from 0 to *RLMultiDecodeNumCompatiblePFK* – 1.

If *TuneAwayStatus* is equal to '1', then the access terminal shall not send any Lower MAC packets.

When an access terminal's RL-ATA changes (and is non-empty), the access terminal shall terminate transmission of the existing packet and shall restart transmission of this packet on the new ATA, unless one of the following two conditions is satisfied:

- a) The RL_ATA change is due to a resource reassignment
- b) The RL-ATA changes due to a HARQ extension

If a positive ACK signal corresponding to the RL-ATA is received (via the FLCS MAC Protocol) that corresponds to the transmitted packet, transmission of the Lower MAC packet shall terminate and transmission of the next packet can begin. If the access terminal transmits a packet for the maximum number of transmissions of the PF selected for this Lower MAC packet, the access terminal shall automatically terminate transmission of the packet. If no ACK is received for this packet after the final HARQ retransmission for that packet, the access terminal shall expire its RL-ATA, and return a *ReverseTrafficPacketsMissed* indication along with parameters that uniquely identify the lost packet. If the access terminal is transmitting a packet, and the RL-ATA used for this packet is expired before an ACK is received, and before the maximum number of transmissions for the PF selected for this Lower MAC packet have occurred, the access terminal shall return a *ReverseTrafficPacketsMissed* indication along with parameters that uniquely identify the lost packet. The method of uniquely identifying the packet is out of the scope of this standard.

If no packet is available for transmission in a given PHY frame, then the access terminal may transmit an erasure sequence in the hop-ports, or subset of hop-ports, assigned to the access terminal in that PHY frame as specified in the subclause on Physical Layer specification. The power for transmission of an erasure sequence is computed as specified in 8.8.5.5.1.9.

8.8.5.5.1.7 Access terminal transmission logic for non-persistent assignments

The access terminal may formulate and transmit a Lower MAC packet on a PHY frame according to the RL-NP-ATA for that PHY frame and the PF selected for transmission. The Lower MAC shall pass the Physical Layer Protocol the set of hop-ports specified by the RL-NP-ATA. The power for transmission of a particular Lower MAC packet is computed as specified in 8.8.5.5.1.9. The access network shall inform the access terminal of its assignment using signaling messages as described in the FLCS MAC Protocol specification.

If the packet format for the RL-NP-ATA matches one of the configuration attributes RLMultiDecodeAssignedPFK, and if the size of the RL-NP-ATA in base nodes is equal to configuration attribute RLMultiDecodeAssignmentSizeK, then the access terminal may transmit one of multiple packet formats, as specified by configuration attributes RLMultiDecodeCompatiblePFKj where j goes from 0 to RLMultiDecodeNumCompatiblePFK – 1.

The number of HARQ retransmissions over a RL-NP-ATA is equal to the maximum number of HARQ retransmissions unless the RL-NP-ATA is extended via an NP-RLAB/NP-RLABs with NULL packet format and supplemental bit '1'.

If TuneAwayStatus is equal to '1', then the access terminal shall not send any Lower MAC packets.

If a positive ACK signal corresponding to the RL-NP-ATA is received (via the FLCS MAC Protocol) that corresponds to the transmitted packet, transmission of the Lower MAC packet shall terminate, and the access terminal shall expire the RL-NP-ATA. If no ACK is received for this packet after the specified number of HARQ retransmissions, the access terminal shall return a *ReverseTrafficPacketsMissed* indication along with parameters that uniquely identify the lost packet. If the access terminal is transmitting a packet, and the RL-NP-ATA used for this packet is expired before an ACK is received, and before the maximum number of retransmissions have occurred, the access terminal shall return a *ReverseTrafficPacketsMissed* indication along with parameters that uniquely identify the lost packet. The method of uniquely identifying the packet is out of the scope of this standard.

8.8.5.5.1.8 Access terminal transmission logic for CDMA Traffic on R-CDCH

For CDMA traffic transmission on R-CDCH, the access terminal shall first determine the value of a parameter, RABObserved, based on the transmissions on F-RABCH from its active set members. If RABEnabled, which is public data of OMP, is set to '0', then the access terminal shall set the RABObserved parameter to '0'. Otherwise, at the beginning of each FL PHY frame, if DuplexingMode = FDD or if

DuplexingMode = TDD44 and each even-indexed FL PHY frame if DuplexingMode = TDD63, the access terminal shall first form a subset of its active set members, denoted by RABMonitorSet, by including those active set members whose ChanDiff value, as defined in 8.8.5.5.1.9.1, is smaller than a threshold, RABMonitorThreshold, where RABMonitorThreshold is a configuration attribute of the RTC MAC protocol. Then, the access terminal shall set the value of RABObserved to '1' if any member of the RABMonitorSet sends a reverse activity indication bit on F-RABCH. Otherwise, the access terminal shall set the RABObserved parameter to '0'.

If the RABObserved parameter is set to '0', then the access terminal may formulate a Lower MAC packet from those flows for which the CDMATransmissionIsAllowedAttribute NN attribute is set to '1' and transmit on R-CDCH. If the RABObserved parameter is set to '1', then the access terminal may formulate a Lower MAC packet from those flows for which the CDMATransmissionIsAllowedAttribute NN attribute is set to '1' and CDMATransmissionRespondsToAttribute NN is set to '0', and transmit on R-CDCH.

R-CDCH transmissions may happen on one or more CDMA subsegments over three consecutive PHY frames starting at PHY frame k , if $k \bmod N_{\text{RLInterlace}}$ belongs to the set $\{\text{PacketStartInterlace}_1, \text{PacketStartInterlace}_2\}$, where $\text{PacketStartInterlace}_1$ and $\text{PacketStartInterlace}_2$ are public data of ASMP. Access terminal's transmissions on RCDCH shall be confined to the CDMA subsegments that are assigned to the access terminal through NumCDCHSubsegments and CDCHSubsegment j parameters, which are public data of the ASMP. For each CDMA subsegment assigned to R-CDCH, the access terminal shall determine the subsegment coordinate (i.e., the interlace index and the CDMA subsegment index on that interlace) using the procedures described in 8.7.5.4.7.

For data transmission on R-CDCH, the access terminal shall use one of the two packet formats described in Table 168.

Table 168—CDMA Traffic packet formats

Packet format index	Packet size in bits	Modulation order for each transmission					
		1	2	3	4	5	6
0	136	2	2	2	2	2	2
1	272	2	2	2	2	2	2

For R-CDCH transmission on PHY frame k , the RTC MAC protocol shall pass the Physical Layer Protocol the set of CDMA subsegment indices on that PHY frame that the access terminal can use for R-CDCH transmission. The set of CDMA subsegment indices shall be constructed from the second entries in the subsegment coordinates of CDCH subsegments for which the first entry is equal to $k \bmod N_{\text{RLInterlace}}$. If the number of such CDMA subsegments is larger than MaxCDCHSubsegmentsPerInterlace, which is a configuration attribute of this protocol, then the set of CDMA subsegment indices shall include only MaxCDCHSubsegmentsPerInterlace subsegments with the smallest indices. The power for transmission of a particular Lower MAC packet over RCDCH is computed as specified in 8.8.5.5.1.10.

The total number of HARQ transmissions allowed for any Lower MAC packet transmitted over RCDCH is 6. HARQ retransmission interval for R-CDCH transmissions is $N_{\text{RLInterlace}}$ RL PHY frames. For a CDMA data packet transmission (or retransmission) starting at RL PHY frame k , an acknowledgment (generated by the SCS MAC protocol) may be transmitted on FL PHY frame k_a . If DuplexingMode = FDD then $k_a = k + 5$, if DuplexingMode = TDD44 then $k_a = k + 4$, and if DuplexingMode = TDD63 then $k_a = 2k + 7$. If a positive ACK signal corresponding to a CDMA traffic packet is received (via the SCS MAC Protocol), transmission of the Lower MAC packet shall terminate. If no ACK is received for this packet after the specified number of HARQ retransmissions, the access terminal shall return a *ReverseTrafficPacketsMissed*

indication along with parameters that uniquely identify the lost packet. The method of uniquely identifying the packet is out of the scope of this standard.

If TuneAwayStatus is equal to '1', then the access terminal shall not send any CDMA traffic packets.

8.8.5.5.1.9 R-ODCH power control

During the transmission of the reverse link data on OFDMA segment, the power per modulation symbol of the R-ODCH on subzone s , $P_{ODCH,s}$, shall be

$$P_{ODCH,s} = P_{PICH} + \Delta_{tx,i,s} + AttemptBoost_j$$

where

- P_{PICH} is the transmit power per modulation symbol for R-PICH and is given in 8.7.5.4.6.3
- i and s are the interlace and subzone indices for the current transmission
- $\Delta_{tx,i,s}$ is as specified in 8.8.5.5.1.9.2
- $AttemptBoost$ is equal to $AttemptBoost_j$ where j is the HARQ transmission index of the current transmission if the $AttemptBoostAllowed/NV$ attribute of all the traffic flows that the access terminal has included in the current packet is '1', and is equal to 0 otherwise

Throughout this subclause, an RL interlace i is defined as the set of all RL PHY frames with index k where $k \bmod N_{RLInterlace} = i$, where $N_{RLInterlace}$ equals 8 and 4 for FDD and TDD, respectively. The total R-ODCH power shall further be subject to the access terminal's transmit power limitation and each $P_{ODCH,s}$ value shall remain constant for the entire transmission of each PHY frame.

If no packet is available for transmission in a given PHY frame, then an erasure sequence shall be transmitted with the following power per modulation symbol:

$$P_{ERASURE,s} = P_{PICH} - pCoT_{RLSS} + InterferenceOffset_{RLSS,s} + ErasureTargetCtoI_i$$

where

- s is the subzone over which the erasure sequence is transmitted
- $InterferenceOffset_{RLSS,s}$ is as defined in 8.7.5.4.6.2
- $ErasureTargetCtoI_i$ $i = 0,1,2$ are configuration attributes of this protocol

Let N_c denote the number of hop-ports in the access terminal assignment. $ErasureTargetCtoI_0$ is used when $N_c = 16$. $ErasureTargetCtoI_1$ is used when $16 < N_c \leq 64$. $ErasureTargetCtoI_2$ is used when $N_c > 64$.

8.8.5.5.1.9.1 FastOSIMonitorSet Update

If the FastOSIEnabled bit is set to '1', and the access terminal had an R-ODCH transmission on RL PHY frame that overlaps with the RL PHY frame k of that sector, then the access terminal shall monitor F-FOSICH transmissions from that sector on all the FL PHY frames that overlap with the FL PHY frame k_a of that sector. If DuplexingMode = FDD then $k_a = k + 5$, if DuplexingMode = TDD44 then $k_a = k + 4$, and if DuplexingMode = TDD63 then $k_a = 2k + 7$. As a result, for a given R-ODCH transmission, the access terminal may be required to monitor multiple F-FOSICH transmissions from a sector that is not mutually synchronous with the RLSS.

On every FL PHY frame, the access terminal shall update the FastOSIMonitorSet with a list of PilotIDs of the sectors whose F-FOSICH transmissions are being monitored by the access terminal on that frame, and whose ChanDiff values, defined as

$$\text{ChanDiff}_s = \frac{\text{RxPower}_{\text{RLSS}}}{\text{EffectiveTransmitPower}_{\text{RLSS}}} \times \frac{\text{EffectiveTransmitPower}_s}{\text{RxPower}_s}$$

are smaller than or equal to FastOSIMonitorThreshold, where FastOSIMonitorThreshold is a configuration attribute of the protocol. In the above definition, s denotes the sector for which the ChanDiff value is being computed, $\text{RxPower}_{\text{RLSS}}$ and RxPower_s are Reference Received Power Levels (as defined by the Physical Layer Protocol) corresponding to RLSS and sector s , and $\text{EffectiveTransmitPower}_{\text{RLSS}}$ and $\text{EffectiveTransmitPower}_s$ are public data of the Overhead Messages Protocol. If the EffectiveTransmitPower value from a sector is not known at the access terminal, the access terminal shall assume that EffectiveTransmitPower for that sector is equal to the DefaultEffectiveTransmitPower value, which is a configuration attribute of this protocol.

The FastOSIMonitorSet shall exclude the PilotID of the RLSS. In addition, if the size of the list is larger than FastOSIMonitorSetSize (where FastOSIMonitorSetSize is a configuration attribute of this protocol), only FastOSIMonitorSetSize PilotIDs with smallest ChanDiff shall be kept.

8.8.5.5.1.9.2 Delta_{tx} determination

If an RLAB is received from FLCS MAC Protocol, which results in a non-empty RL-ATA, then the access terminal shall interpret the RLPC field in the RLAB as follows:

- The value 31 shall be treated as the reserved value.
- Values between 0 and 30 shall be interpreted as explicit DataCtoI assignment according to the following expression:

$$\text{DataCtoI}_{\text{assigned}} = \text{DataCtoI}_{\text{Nom},p} + \text{RLPCFieldValue} - 15,$$

where p is the index of the assigned packet format, $\text{DataCtoI}_{\text{Nom},p}$ is the nominal (reference) DataCtoI value associate with this packet format as given in Table 168, and RLPCFieldValue denotes the value of the RLPC field in the RLAB.

If the RLPC field in the RLAB is not the reserved value, then before the first transmission of the new assignment, the access terminal shall set the $\text{DataCtoI}_{\text{min}}$, Δ_{tx} , and Δ_{max} values according to the following rules:

$$\text{DataCtoI}_{\text{min},i} = \text{DataCtoI}_{\text{assigned}} - \text{MaxCtoIReduction}$$

$$\Delta_{\text{tx},i,s} = \text{InterferenceOffset}_{\text{RLSS},s} - \text{pCoT}_{\text{RLSS}} + \text{DataCtoI}_{\text{assigned}}$$

$$\Delta_{\text{max},i,s} = \Delta_{\text{tx},i,s} + \text{MaxDeltaIncrease}$$

where MaxCtoIReduction is the largest MaxCtoIReduction/ N value among all the traffic flows that the access terminal will include in the packets transmitted using this assignment, MaxDeltaIncrease is the smallest MaxDeltaIncrease/ N value among all the traffic flows that the access terminal will include in the packets transmitted using this assignment, and $\text{InterferenceOffset}_{\text{RLSS},s}$ is as defined in 8.7.5.4.6.2. Here i and s are the interlace and subzone indices for the current assignment. For assignments that span more than one subzone, the access terminal shall maintain multiple Δ_{tx} , Δ_{max} , and Δ_{min} values corresponding to different subzones in the assignment.

After every update of the $\text{InterferenceOffset}_{\text{RLSS}}$ and $\text{pCoT}_{\text{RLSS}}$ values, the access terminal shall set the Δ_{min} values according to the following rule:

$$\Delta_{\text{min},i,s} = \text{InterferenceOffset}_{\text{RLSS},s} - \text{pCoT}_{\text{RLSS}} + \text{DataCtoI}_{\text{min},i}$$

Here i and s are the interlace and subzone indices for the current assignment. For assignments that span more than one subzone, the access terminal shall maintain multiple Δ_{tx} , Δ_{max} , and Δ_{min} values corresponding to different subzones in the assignment.

If FastOSIEnabled bit is set to '1', then after every FastOSIMonitorSet update, the access terminal shall update the Δ_{tx} values according to the following rules:

$$\Delta_{tx,i,s} = \begin{cases} \Delta_{tx,i,s} + \text{FastOSIStepUp} & \text{if any FastOSI}_{j,s} = 0 \\ \Delta_{tx,i,s} - \text{FastOSIStepDown} & \text{if any FastOSI}_{j,s} = 1 \end{cases}$$

$$\Delta_{tx,i,s} = \max \{ \min \{ \Delta_{tx,i,s}, \Delta_{max,i,s} \}, \Delta_{min,i,s} \}$$

where FastOSIStepUp and FastOSIStepDown are configuration attributes of the protocol.

Here, $\text{FastOSI}_{j,s}$ denotes the fast OSI indication received from the j^{th} member of the FastOSIMonitorSet for subzone s , and i is the interlace index of the RL PHY frame corresponding to the FastOSI values. In the FDD mode, the FastOSI value carried over the FL PHY frame with index k is generated for RL PHY frame with index $k - 5$. In the TDD 1:1 mode, the FastOSI value carried over the FL PHY frame with index k is generated for RL PHY frame with index $k - 4$.

Finally, the access terminal shall restrict the value of Δ_{tx} based on the corresponding Δ_{min} and Δ_{max} values, according to the following rule:

$$\Delta_{tx,i,s} = \max \{ \min \{ \Delta_{tx,i,s}, \Delta_{max,i,s} \}, \Delta_{min,i,s} \}$$

8.8.5.5.1.9.3 OSIMonitorSet update

At the beginning of every superframe of the RLSS, the access terminal shall update the OSIMonitorSet with a list of PilotIDs of the sectors whose PilotStrength is greater than or equal to the OSIMonitorThreshold, where PilotID and PilotStrength are public data of the ASMP and OSIMonitorThreshold is a configuration attribute of this protocol.

The OSIMonitorSet shall exclude the PilotID of the RLSS. In addition, if the size of the list is greater than OSIMonitorSetSize (where OSIMonitorSetSize is a configuration attribute of this protocol), only OSIMonitorSetSize PilotIDs with strongest PilotStrength shall be kept.

8.8.5.5.1.9.4 Δ_{slow} determination

Δ_{slow} , OSI2SequenceNum, and PilotIDStrongest are the access terminal's locally maintained parameters and shall be initialized to DataCoTMin, 1 and -1, respectively.

After every update of the $pCoT_{RLSS}$ value, the access terminal shall set the $\Delta_{slow,min}$ and $\Delta_{slow,max}$ values according to the following rules:

$$\Delta_{slow,min} = \text{DataCoTMin} - pCoT_{RLSS}$$

$$\Delta_{slow,max} = \text{DataCoTMax} - pCoT_{RLSS}$$

where DataCoTMin and DataCoTMax are configuration attributes of this protocol.

After each OSIMonitorSet update, the access terminal shall create an OSI vector whose i^{th} element, i.e., OSI_i , corresponds to the most recent OSIValue from the sector whose PilotID is indicated by the i^{th} entry of

the OSIMonitorSet. The most recent OSIValue is the most recent value received over the FOSICH of the sector. In addition, the access terminal shall create a ChanDiff vector, whose i^{th} element is the ChanDiff value corresponding to the i^{th} element of the OSIMonitorSet.

The access terminal shall determine Δ_{slow} as follows:

If the OSIMonitorSet is empty, the access terminal shall set Δ_{slow} to $\Delta_{\text{slow,max}}$, OSI2SequenceNum to 1 and PilotIDStrongest to -1.

If the OSIMonitorSet is not empty, the access terminal shall compute the Δ_{slow} as follows:

The access terminal shall first compute a Decision Threshold vector, whose i^{th} element, i.e., $\text{DecisionThreshold}_i$, $1 \leq i \leq \text{OSIMonitorSetSize}$, is given by:

$$\text{DecisionThreshold}_i = \begin{cases} \max\{\text{UpDecisionThresholdMin}, (1-a)b_i\} & \text{if } \text{OSI}_i = 0 \\ \max\{\text{DownDecisionThresholdMin}, a(1-b_i)\} & \text{if } \text{OSI}_i = 1 \\ 1 & \text{if } \text{OSI}_i = 2 \end{cases}$$

where UpDecisionThresholdMin and DownDecisionThresholdMin are configuration attributes of the protocol. Variables a and b_i are determined as follows:

$$a = \frac{\min\{\Delta_{\text{slow}}, \Delta_{\text{slow,max}}\} - \Delta_{\text{slow,min}}}{\Delta_{\text{slow,max}} - \Delta_{\text{slow,min}}}, \text{ and}$$

$$b_i = \frac{\min\{\text{ChanDiff}_i, \text{ChanDiffMax}\} - \text{ChanDiffMin}}{\text{ChanDiffMax} - \text{ChanDiffMin}},$$

where ChanDiffMax and ChanDiffMin are configuration attributes of the protocol, and all values in the above computations are in logarithmic scale (in units of dB).

- a) If OSIResponseMode is set to “Stochastic,” then the access terminal shall produce a Decision vector whose i^{th} element, i.e., Decision_i , $1 \leq i \leq \text{OSIMonitorSetSize}$, is given by:

$$\text{Decision}_i = \begin{cases} \text{UpDecisionValue} & \text{if } x_i \leq \text{DecisionThreshold}_i \text{ and } \text{OSI}_i = 0 \\ -\text{DownDecisionValue} & \text{if } x_i \leq \text{DecisionThreshold}_i \text{ and } \text{OSI}_i = 1 \text{ or } 2 \\ 0 & \text{otherwise} \end{cases}$$

where $0 \leq x_i \leq 1$ is a uniform random variable (generated using the procedure specified in 9.2.5) and UpDecisionValue and DownDecisionValue are configuration attributes of the protocol.

- b) If OSIResponseMode is set to “Deterministic,” then the access terminal shall produce a Decision vector whose i^{th} element, i.e., Decision_i , $1 \leq i \leq \text{OSIMonitorSetSize}$, is given by:

$$\text{Decision}_i = \begin{cases} \text{UpDecisionValue} \times \text{DecisionThreshold}_i & \text{if } \text{OSI}_i = 0 \\ -\text{DownDecisionValue} \times \text{DecisionThreshold}_i & \text{if } \text{OSI}_i = 1 \text{ or } 2 \end{cases}$$

- c) The access terminal shall then compute a weighted decision, D_w , according to:

$$D_w = \frac{\sum_{i=1}^{OSIMonitorSetSize} \frac{1}{ChanDiff_i} Decision_i}{\sum_{i=1}^{OSIMonitorSetSize} \frac{1}{ChanDiff_i}}$$

The access terminal shall find the sector with the lowest ChanDiff in the OSIMonitorSet and call that sector as sector k. Then the access terminal shall set the variable OSIStrongest to the OSI value of sector k and PilotIDCurrent to the PilotID of sector k. Then, the access terminal shall update OSI2SequenceNum and PilotIDStrongest as follows:

$$OSI2SequenceNum = \begin{cases} OSI2SequenceNum + 1 & \text{if } PilotPNCCurrent = PilotPNStrongest \text{ and } OSIStrongest = 2 \\ 2 & \text{if } PilotPNCCurrent \neq PilotPNStrongest \text{ and } OSIStrongest = 2 \\ 1 & \text{otherwise} \end{cases}$$

$$PilotPNStrongest = \begin{cases} PilotPNCCurrent, & \text{if } OSIStrongest = 2 \\ -1 & , \text{ otherwise} \end{cases}$$

Furthermore, the OSI2SequenceNum value shall be subject to a maximum value of OSI2SequenceNumMax where OSI2SequenceNumMax is a configuration attribute of the protocol.

- If OSIResponseMode is set to “Stochastic,” then the access terminal shall increase the Δ_{slow} value by OSISStepUp dB if D_w is greater than DwUpThreshold and shall decrease the Δ_{slow} value by OSISStepDown $\times$ OSI2SequenceNum dB if D_w is less than or equal to $-DwDownThreshold$.
- If OSIResponseMode is set to “Deterministic,” then the access terminal shall increase the Δ_{slow} value by OSISStepUp $\times D_w$ dB if D_w is greater than 0 and shall decrease the Δ_{slow} value by OSISStepDown $\times$ OSI2SequenceNum $\times D_w$ dB if D_w is less than or equal to 0.

In the above rules, OSISStepUp, OSISStepDown, DwUpThreshold, and DwDownThreshold are configuration attributes of this protocol. Furthermore, the Δ_{slow} value is constrained to always lie between $\Delta_{slow,min}$ and $\Delta_{slow,max}$. That is, the access terminal shall set Δ_{slow} to $\Delta_{slow,min}$ if the resulting Δ_{slow} was smaller than $\Delta_{slow,min}$ and to $\Delta_{slow,max}$ if the resulting Δ_{slow} was larger than $\Delta_{slow,max}$.

8.8.5.5.1.10 R-CDCH power control

If the number of CDMA subsegments assigned to the access terminal for CDMA traffic transmission, i.e., the value of NumCDCHSubsegments, is not zero, then corresponding to each configuration attribute AuxPICHOffsetNN, the access terminal shall initialize a $P_{AuxPICH,tot}$ value as

$$P_{AuxPICHNN,tot} = P_{PICH} + 10\log_{10}N_{PICH} + AuxPICHOffsetNN,$$

where P_{PICH} is the transmit power per modulation symbol for R-PICH and is given in 8.7.5.4.6.3 and $N_{\text{PICH}} = 1024$ is the total number of modulation symbols per R-PICH transmission.

For each packet that is transmitted over R-CDCH, if a positive acknowledgement is received for that packet on F-ACKCH within the TerminationTarget $_{NN}$ HARQ attempts, the AuxPICHOffset $_{NN}$ value shall be decreased by AuxPICHStepDown $_{NN}$; otherwise, the AuxPICHOffset $_{NN}$ value shall be increased by AuxPICHStepUp $_{NN}$. Here, NN is index of the traffic stream that the access terminal has included in the current packet being transmitted over the R-CDCH, and TerminationTarget $_{NN}$, AuxPICHStepDown $_{NN}$, and AuxPICHStepUp $_{NN}$ are configuration attributes of this protocol. If the packet contains data from multiple traffic streams, then the access terminal shall use the TerminationTarget, AuxPICHOffset, AuxPICHStepUp, and AuxPICHStepDown parameters corresponding to the traffic stream with the earliest TerminationTarget. If multiple streams have the same TerminationTarget, then among all the streams with the earliest TerminationTarget, the access terminal shall choose the stream with the largest value of AuxPICHStepUp/AuxPICHStepDown. In the rest of this subclause, $P_{\text{AuxPICH},\text{tot}}$ will be used to refer to the $P_{\text{AuxPICH},\text{tot}}$ value corresponding to this Stream.

During the transmission of the reverse link data on CDMA segment on PHY frame k , the power per modulation symbol of the R-AuxPICH ($P_{\text{AuxPICH},i}$) and R-CDCH ($P_{\text{CDCH},k}$) shall be:

$$P_{\text{AuxPICH},k} = P_{\text{AuxPICH},\text{tot}} - 10\log_{10}(N_{\text{PHY-FRAMES}}) - 10\log_{10}(N_{\text{OFDM-SYMBOLS}}N_{\text{CDCH-SUBCARRIERS}})$$

$$P_{\text{CDCH},k} = P_{\text{AuxPICH},k} + \text{CDMATrafficPilotOffset}_p$$

where

- $N_{\text{OFDM-SYMBOLS}} = 8$ is the number of OFDM symbols in each regular RL PHY frames
- $N_{\text{CDCH-SUBCARRIERS}}$ is the number subcarriers allocated for R-CDCH on PHY frame k and is equal to $128 \times N_{\text{CDCH-SUBSEGMENTS}}$
- p is the index of the packet format used for this transmission
- $\text{CDMATrafficPilotOffset}_p$ is a configuration attribute of this protocol

$N_{\text{CDCH-SUBSEGMENTS}}$ and $N_{\text{PHY-FRAMES}}$ are determined according to the following rules:

- First, a start of packet interlace, s , shall be determined by selecting the element of the set $\{\text{mod}(k-2, N_{\text{RLInterlace}}), \text{mod}(k-1, N_{\text{RLInterlace}}), \text{mod}(k, N_{\text{RLInterlace}})\}$ that also belongs to the set $\{\text{PacketStartInterlace}_1, \text{PacketStartInterlace}_2\}$, where $\text{PacketStartInterlace}_1$ and $\text{PacketStartInterlace}_2$ are public data of ASMP.
- Then, the $N_{\text{PHY-FRAMES}}$ value shall be set to the number of interlaces in the set $\{s, \text{mod}(s+1, N_{\text{RLInterlace}}), \text{mod}(s+2, N_{\text{RLInterlace}})\}$ that are assigned to the access terminal as CDCH interlaces. An interlace i is said to be assigned to the access terminal as CDCH interlace if there is a subsegment coordinate, as defined in 8.7.5.4.7, corresponding to at least one of the CDCHSubsegment values assigned to the access terminal whose first entry is equal to i .
- Next, the parameter $N_{\text{CDCH-SUBSEGMENTS}}$ shall be set to the minimum of $\text{MaxCDCHSubsegmentsFrame}$ and the number of subsegment coordinates corresponding to CDCHSubsegment indices assigned to the access terminal whose first entry is equal to $k \bmod N_{\text{RLInterlace}}$.

The total R-AuxPICH and R-CDCH power shall further be subject to the access terminal's transmit power limitation and the $P_{\text{AuxPICH},i}$ and $P_{\text{CDCH},i}$ values shall remain constant for the entire transmission of each PHY frame.

8.8.5.5.2 Access network requirements for persistent assignments

For each set of PHY frames with a non-empty RL-ATA (contains one or more hop-ports), the access network may attempt to decode a Lower MAC packet. The access network may attempt to detect erasure sequences that are transmitted by the access terminal whenever a Lower MAC packet is not available for transmission. Exact algorithms for detecting erasure sequences and the start-of-packet for Lower MAC packets that span multiple PHY frames are beyond the scope of this standard.

If the packet format for the RL-ATA matches one of the configuration attributes `RLMultiDecodeAssignedPFK`, and if the size of the RL-ATA in base nodes is equal to configuration attribute `RLMultiDecodeAssignmentSizeK`, then the access network will try to decode multiple packet formats, as specified by configuration attributes `RLMultiDecodeCompatiblePFKj` where j goes from 0 to `RLMultiDecodeNumCompatiblePFK` – 1.

If `TuneAwayStatus` is equal to '1', then the access network should not attempt to decode any packets from that access terminal.

If `RLImplicitDeassignEnabled` is equal to '1', then upon receiving a *TunedAway* indication from the Air link Management Protocol, the access network shall expire any RL-ATAs for that access terminal.

If a Lower MAC packet is successfully decoded, as indicated by the PHY, the access network shall transmit a positive ACK on the F-SCCH channel via the FLCS MAC Protocol. If a Lower MAC packet fails to decode and the access network determines that the packet has been transmitted for the total number of PHY frames for that packet (and relevant PF), then the access network shall expire the RL-ATA. Refer to 8.1.5.2 for detailed timing structure and acknowledgment timing.

Exact algorithms to determine the number of HARQ attempts prior to successful decode are beyond the scope of this standard.

If the `RLNumSDMADimensions` > 1, then the access network shall ensure that assignments to different access terminals that contain hop-ports that map to the same subcarriers have the same F-DPICH format, as described in 8.8.5.10.

8.8.5.5.3 Access network requirements for non-persistent assignments

For each set of PHY frames with a non-empty RL-NP-ATA (contains one or more hop-ports), the access network may attempt to decode a Lower MAC packet.

If the packet format for the RL-NP-ATA matches one of the configuration attributes `RLMultiDecodeAssignedPFK`, and if the size of the RL-NP-ATA in base nodes is equal to configuration attribute `RLMultiDecodeAssignmentSizeK`, then the access network will try to decode multiple packet formats, as specified by configuration attributes `RLMultiDecodeCompatiblePFKj` where j goes from 0 to `RLMultiDecodeNumCompatiblePFK` – 1.

If `TuneAwayStatus` is equal to '1', then the access network should not attempt to decode any packets from that access terminal.

If `RLImplicitDeassignEnabled` is equal to '1', then upon receiving a *TunedAway* indication from the Air link Management Protocol, the access network shall expire any RL-NP-ATAs for that access terminal.

If a Lower MAC packet is successfully decoded, as indicated by the PHY, the access network shall transmit a positive ACK on the F-SCCH channel via the FLCS MAC Protocol, and shall expire the RL-NP-ATA.

Exact algorithms to determine the number of HARQ attempts prior to successful decode are beyond the scope of this standard.

If the $RLNumSDMADimensions > 1$, then the access network shall ensure that (non-persistent) assignments to different access terminals that contain hop-ports that map to the same subcarriers have the same FDPICH format, as described in 8.8.5.10.

The RL-NP-ATA shall expire after the total number of HARQ retransmissions for that ATA if it is not already expired.

8.8.5.6 Reverse link Overhead UpToDate enforcement

The access terminal shall not transmit on any RL channel to a particular sector if that sector has its SectorChannelInfoUpToDate public data of the Overhead Messages protocol equal to 0.

8.8.5.7 Parameter management and message processing

The access terminal shall maintain a parameter Override for each sector in the Active Set. Upon addition of a sector to the Active Set, the access terminal shall initialize Override = NULL. If the access terminal receives a TrafficPowerControlOverride message for a sector, it shall set Override = NON-NULL for that sector.

If Override = NULL, the access terminal shall set the parameters AttemptBoost_j, AttemptBoostAllowed_{NN}, MaxCtoIReduction_{NN}, MaxDeltaIncrease_{NN}, TerminationTarget_{NN}, AuxPICHOffset_{NN}, CDMATrafficPilotOffset_p, AuxPICHStepDown_{NN}, and AuxPICHStepUp_{NN} to AttemptBoostAttribute_j, AttemptBoostAllowedAttribute_{NN}, MaxCtoIReductionAttribute_{NN}, MaxDeltaIncreaseAttribute_{NN}, TerminationTargetAttribute_{NN}, AuxPICHOffsetAttribute_{NN}, CDMATrafficPilotOffsetAttribute_p, AuxPICHStepDownAttribute_{NN}, and AuxPICHStepUpAttribute_{NN}, respectively.

If Override = NON-NULL, the access terminal shall set AttemptBoost_j, AttemptBoostAllowed_{NN}, MaxCtoIReduction_{NN}, MaxDeltaIncrease_{NN}, TerminationTarget_{NN}, AuxPICHOffset_{NN}, CDMATrafficPilotOffset_p, AuxPICHStepDown_{NN}, and AuxPICHStepUp_{NN} to the values in the last received override message.

8.8.5.8 Supervision procedures

The access terminal shall generate a *SupervisionFailed* indication when it has an empty RL-ATA during a time period of length $T_{RTCSupervision}$ while it has sent non-empty request bits during all R-REQCH transmissions during the time period.

8.8.5.9 Channel trees

A channel tree defines the mapping of each NodeID to a set of hop-ports and the grouping of hop-ports into port-sets. The number of subcarriers mapped by the channel tree, N_{FFT} , a parameter that is defined by the Physical Layer protocol. See 8.6.5.6 for common terms used for describing channel trees in this standard. Hop-ports shall be numbered numerically from 0.

Because nodes define orthogonal assignments, the use of a node in the tree can restrict use of other nodes. Thus, if a node is in use, then all descendants and ancestors of the node are unavailable for use and are called “restricted” nodes.

The channel tree is described as a function of the variable N_{FFT} , which is given by the TotalNumSubcarriers field of the OMP.

The set of hop-ports specified by a NodeID is given by the union of usable hop-ports mapped by all base-nodes that are descendants of the node specified by NodeID. “Usable hop-ports” are defined by the Physical Layer Protocol.

In the case of SDMA transmission, multiple modulation symbols are multiplexed over the same physical subcarriers at different spatial dimensions. For each spatial dimension, an SDMA subtree is define to specify all the associated hop-ports. The SDMA order for reverse link transmissions, denoted by Q_{SDMA} , is equal to $\text{RLNumSDMADimensions}$, which is public data of the OMP. The number of hop-ports indexed by a channel tree shall equal $Q_{\text{SDMA}}N_{\text{FFT}}$.

The set of $Q_{\text{SDMA}}N_{\text{FFT}}$ hop-ports is divided into Q_{SDMA} “SDMA subtrees,” indexed 0 through $Q_{\text{SDMA}} - 1$. Each SDMA subtree has N_{FFT} hop-ports, and the SDMA subtree with index q contains hop-ports indexed qN_{FFT} through $(q + 1)N_{\text{FFT}} - 1$.

SDMA subtree with index 0 is illustrated in Figure 62 for $N_{\text{FFT}} = 512, 1024$, and 2048. For $N_{\text{FFT}} = X$, all nodes above the dashed line marked with “ $N_{\text{FFT}} = X$ ” are included in the channel tree. One deassignment NodeID, $\text{NodeID}_{\text{DEASSIGN}}$, is set to $N_{\text{FFT}}/(\text{MinHopPortsPerNode}/2) - 1$. Other SDMA subtrees are identical as the SDMA subtree 0, but with unique NodeID to hop-ports mapping.

The mapping of hop-ports to each base-node is described as follows. Each base-node maps to $\text{MinHopPortsPerNode}$ hop-ports, the first $\text{MinHopPortsPerNode}$ hop-ports (indices 0 to $\text{MinHopPortsPerNode}-1$) to the base-node with the lowest NodeID, the second $\text{MinHopPortsPerNode}$ hop-ports to the next base-node, etc., until all hop-ports are mapped. See Table 169 for an example of this mapping for $N_{\text{FFT}} = 512$, $\text{MinHopPortsPerNode} = 16$, and $Q_{\text{SDMA}} = 4$.

**Table 169—RL base node NodeID and subtree index to hop-port mapping example
($N_{\text{FFT}} = 512$, $\text{MinHopPortsPerNode} = 16$, and $Q_{\text{SDMA}} = 4$)**

(Base node NodeID, subtree index)	Hop-ports mapped
(31, 0)	0–15
(32, 0)	16–31
...	...
(62, 0)	496–511
(31, 1)	512–527
(32, 1)	528–543
...	...
(62, 1)	1008–1023
...	...
(62, 3)	2032–2047

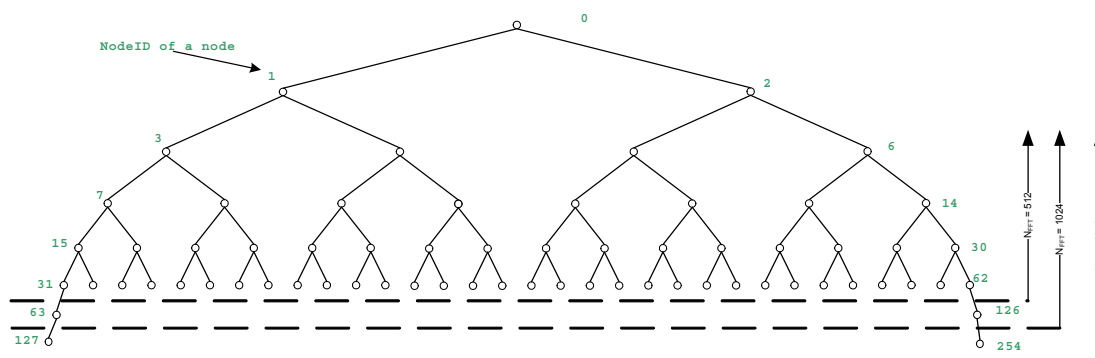


Figure 62—Reverse Link SDMA subtree Index 0

8.8.5.10 Packet formats

A packet format (PF) specifies the spectral efficiency, maximum number of transmissions, and the modulation format to be used for each transmission of a data packet. The packet format consists of four bits. These four bits index the spectral efficiency, maximum number of transmissions, and the modulation format to be used for each transmission of a data packet. The indexing is described in Table 170.²⁴

The modulation format is specified by the number of bits in each modulation symbol, which is denoted by modulation order. Modulation orders of 2, 3, 4, and 6 correspond to QPSK, 8PSK, 16-QAM, and 64-QAM modulations, respectively. The size of the Lower MAC packet that is provided to the Physical Layer is a function of the packet format as well as the set of hop-ports that are assigned to the data packet (to be transmitted on the R-DCH Physical Layer channel). The packet size is byte aligned. Also, a CRC of length $N_{\text{CRC,Data}}$ is added by the Physical Layer for F-DCH packets, where $N_{\text{CRC,Data}}$ is a constant of the Physical Layer.

Note that the default maximum number of transmissions is six; however, a persistent assignment could be extended to more than six transmissions through a RLAB message.

The packet size for each assignment is computed based on the spectral efficiency listed in Table 170 and the assignment size. The packet size is also byte aligned. Let s denote the spectral efficiency of the packet format at the first HARQ transmission (specified in the packet format table); let H denote the number of usable hop-ports in the assignment at the first HARQ transmission; let N denote the number of information bits in the packet excluding CRC. Then, the packet size is given by

$$N = 8 \lfloor sH \rfloor - N_{\text{CRC,Data}}$$

for regular assignments, and

$$N = 8 \lfloor 3sH \rfloor - N_{\text{CRC,Data}}$$

for extended frame assignments, respectively.

²⁴For the same modulation format, different spectral efficiencies are achieved by using a different code rate, i.e., by transmitting different numbers of redundant bits.

Table 170—Reverse link packet formats

Packet format index	Spectral efficiency on first transmission	Reference (Nominal) DataCtol in dB	Modulation order for each transmission					
			1	2	3	4	5	6+
0	0.44	−1	2	2	2	2	2	2
1	0.71	1	2	2	2	2	2	2
2	1.07	2	2	2	2	2	2	2
3	1.4	4	3	2	2	2	2	2
4	1.8	5	4	3	3	3	3	3
5	2.13	7.5	6	4	4	4	4	4
6	2.5	7.5	6	4	4	4	4	4
7	3.0	8.5	6	4	4	4	4	4
8	3.5	10	6	4	4	4	4	4
9	4.0	10.5	6	6	4	4	4	4
10	4.5	11.5	6	6	4	4	4	4
11	5.0	13	6	6	4	4	4	4
12	6.0	14	6	6	4	4	4	4
13	7.0	16	6	6	6	4	4	4
14	8.0	17	6	6	6	4	4	4
15	NULL	—	—	—	—	—	—	—

8.8.6 Header and trailer and formats

The access terminal shall formulate a packet for transmission over the Reverse Traffic Channel using the following header and trailer.

8.8.6.1 Trailer

An InBand Trailer (see Table 171) may be included by this protocol if there is space in the packet left over after all Packet Consolidation Protocol packets have been fitted into the available Lower MAC packet size.

- If no space is left over, the trailer shall not be included.
- The all zero trailer is reserved, and shall be used to indicate the end of Lower MAC Trailers.

Table 171—RL MAC trailer

Field	Length (bits)
RLInBandControl	0 or $N \times 8$

RLInBandControl The RL in-band bits are used to transmit power control, buffer level, and packet latency information to the access network RL scheduler. The RLInBandControl blocks are given in 8.8.6.3. The number of InBand trailers depends on the available space in the Lower MAC Packet after all PCP Packets have been accommodated.

8.8.6.2 Header

This protocol does not specify a header.

8.8.6.3 RLInBandControl

There are four possible RLInBandControl blocks listed in 8.8.6.3.1 through 8.8.6.3.4. Multiple RLInBandControl blocks may be sent in one Lower MAC Header.

8.8.6.3.1 InBandPAH Block

The first RLInBandControl block, shown in Table 172, transmits RL power amplifier headroom information. The access terminal shall include 0 or one InBandPAH block in a Lower MAC packet.

When the PAHeadroom changes by more than MinPAHChange since the last sent in-band or out-of-band report, the access terminal shall start a timer with duration InBandPAHTimeout frames. If the access terminal starts any packet transmission while the timer is running, the access terminal shall include an InBand report and disable the timer. If the timer expires, the access terminal may send the report using the out-of-band channel. MinPAHChange is a configuration attribute of the RCC MAC Protocol and InBandPAHTimeout is a configuration attribute of this protocol.

Table 172—InBandPAH block format

Field	Length (bits)
BlockFormat	2
PAHeadroom	6

BlockFormat The access terminal shall indicate an RL PA headroom block by setting the format indicator bits to '00'.

PAHeadroom PAHeadroom is a quantized value of the maximum achievable CoT value over the entire band, where the CoT is defined as the average carrier power divided by the average thermal power in the band in dB, that is $\text{CoT} = 10\log_{10}(\text{Carrier Power}/\text{Thermal Power})$. The rule for determining the PAHCHValue is specified in 8.7.5.4.18.3. The 6 bits in the PAHeadroom field indicate values in the range -26 dB to 26 dB as specified in Table 173. The access terminal shall report the value from this table that is closest to the observed value.

Table 173—PAHeadroom in dB

PAHeadroom	Value in dB
1–31	-26 to -11 in steps of 0.5
32–57	-10 to 15 in steps of 1
58–63	16 to 26 in steps of 2

Note that the all 0 value is reserved for zero padding.

8.8.6.3.2 InBandREQ Block

The second RLInBandControl block, shown in Table 174, transmits RL QoS class and status of buffer at the access terminal.

The access terminal's inclusion of InBandREQ does not have any requirements if MinInBandReqRequiredPacketSize is equal to 0x00.

Otherwise, The access terminal is required to include an InBandREQ block when the access terminal has an R-ODCH assignment of size at least MinInBandReqRequiredpacketSize bytes, and the access terminal has unsent data in the buffer.

A InBand request timer shall be said to be active if any of the following conditions is satisfied

- The access terminal completed transmission of a InBand request within the last InBandRequestTimer PHY frames, or
- The access terminal is currently transmitting a packet containing an InBand request and InBandRequestTimer is not equal to zero.

Table 174—InBandREQ block format

Field	Length (bits)
BlockFormat	2
REQReport	6

BlockFormat The access terminal shall indicate an RL REQ block by setting the format indicator bits to '01'.

REQCHReport The access terminal shall specify the REQReport according to 8.7.5.4.17.2.

8.8.6.3.3 InBandDataCtoi Block

The third RLInBandControl block, shown in Table 175, is used to transmit DataCtoi value corresponding to the current Δ_{tx} value on the interlace over which the current packet is being transmitted.

The rules for determining the Δ_{tx} value are explained in 8.8.5.5.1.9.2. The DataCtoi value corresponding to the current Δ_{tx} value on subzone s shall be computed as

$$\text{DataCtoi}_s = \Delta_{tx,s} - \text{InterferenceOffset}_{RLSS,s} + pCoT_{RLSS}$$

If the current assignment of the access terminal spans more than one subzone, the access terminal shall compute a single DataCtoi value by averaging all the DataCtoi values on the subzones occupied by this assignment. If the Δ_{tx} value is being reset at the beginning of the current packet by an explicit DataCtoi_{assigned} through the RLPC field of an RLAB, the access terminal shall report the DataCtoi value corresponding to the Δ_{tx} value before reset.

When the DataCtoi changes by more than MinDataCtoiChange since the last sent in band report, an InBandDataCtoi block should be included in the next packet transmission. MinDataCtoiChange is a parameter defined in the Configuration attributes of this protocol.

Table 175—InBandDataCtoI Block

Field	Length (bits)
BlockFormat	3
DataCtoI	5

BlockFormat The access terminal shall indicate an RL DataCtoI block by setting the format indicator bits to '100'.

DataCtoI DataCtoI is defined as the data CtoI for the transmission of current interlace averaged over assigned subzones, as explained above. The quantization is defined in Table 176.

Table 176—DataCtoI in dB

DataCtoI	Value in dB
0–30	–5 to 25 in steps of 1
31	reserved

8.8.6.3.4 InBandDeltaSlow Block

The fourth RLInBandControl block, shown in Table 177, is used to transmit the DataCoT value corresponding to the current Δ_{slow} value. The rules for determining the Δ_{slow} value are specified in 8.8.5.5.1.9.4. The DataCoT value corresponding to the current Δ_{slow} value shall be computed as

$$\text{DataCoT} = \Delta_{\text{slow}} + \text{pCoT}_{\text{RLSS}},$$

where $\text{pCoT}_{\text{RLSS}}$ is the most recent pCoT value corresponding to the RLSS.

When the DataCoT value changes by more than MinDeltaSlowChange since the last sent in band report, an InBandDeltaSlow block should be included in the next packet transmission. MinDeltaSlowChange is a parameter defined in the Configuration attributes of this protocol.

Table 177—InBandDeltaSlow block format

Field	Length (bits)
BlockFormat	3
DataCoT	5

BlockFormat The access terminal shall indicate an RL DataCoT block by setting the format indicator bits to '101'.

DataCoT The DataCoT value, limited to the range of DataCoTMin to DataCoTMax (see Table 178).

Table 178—DeltaSlow in dB

DataCoT	Value in dB
0–30	Offset in steps of $(\text{DataCoTMax} - \text{DataCoTMin})/30$ relative to DataCoTMin
31	Reserved

8.8.7 Message formats

8.8.7.1 TrafficPowerControlOverride

This message may be used by the access network to reset the AttemptBoost_j, AttemptBoostAllowed_{NN}, MaxCtoIReduction_{NN}, MaxDeltaIncrease_{NN}, TerminationTarget_{NN}, CDMATrafficPilotOffset, AuxPICHOffset_{NN}, AuxPICHStepDown_{NN}, and AuxPICHStepUp_{NN} attributes at the access terminal for a particular pilot in the active set. (See Table 179.)

Table 179—TrafficPowerControlOverride message format

Field	Length (bits)
MessageID	8
PilotID	10
AttemptBoostOverride _j , $j = 1, \dots, 6$	6x4
CDMATrafficPilotOffsetOverride _p , $p=0,1$	2x2
NumStreams	6
NumStreams instances of the following six fields {	
<i>NN</i>	8
AttemptBoostAllowedOverride _{NN}	1
MaxCtoIReductionOverride _{NN}	4
MaxDeltaIncreaseOverride _{NN}	4
TerminationTargetOverride _{NN}	3
AuxPICHOffsetOverride _{NN}	3
AuxPICHStepDownOverride _{NN}	8
AuxPICHStepUpOverride _{NN}	8
}	
Padding	Variable

MessageID	The access network shall set this field to 0x00.
PilotID	The access network shall set this field to the PilotID corresponding to the sector for which this message is valid.
AttemptBoostOverride _j	The new value for AttemptBoost _j in units of dB.
CDMATrafficPilotOffsetOverride _p	The new value for CDMATrafficPilotOffset _p .
NumStreams	The number of streams for which the power control parameters are being altered
<i>NN</i>	The stream to which the next two fields correspond.
AttemptBoostAllowedOverride _{NN}	The new value for AttemptBoostAllowed _{NN} .

MaxCtoIReductionOverride*NN*
The new value for MaxCtoIReduction*NN*, in units of 1 dB.

MaxDeltaIncreaseOverride*NN*
The new value for MaxDeltaIncrease*NN*, in units of 1 dB.

TerminationTargetOverride*NN*
The new value for TerminationTarget*NN*.

AuxPICHOffsetOverride*NN*
The new value for AuxPICHOffset*NN*.

AuxPICHStepDownOverride*NN*
The new value for AuxPICHStepDown*NN*, in units of 0.01 dB.

AuxPICHStepUpOverride*NN*
The new value for AuxPICHStepUp*NN*, in units of 0.01 dB.

Padding
The length of this field shall be such that this message is octet aligned. The sender shall set this field to zero. The receiver shall ignore this field.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

8.8.8 Interface to other protocols

8.8.8.1 Commands

This protocol issues the following commands:

- *ReverseControlChannelMAC.ChangeFLSS*
- *ReverseControlChannelMAC.ChangeRLSS*
- *ReverseControlChannelMAC.ACKAssignment*

8.8.8.2 Indications

This protocol registers to receive the following indications:

- *AirLinkManagement.TunedAway*
- *ReverseControlChannelMAC.RLSSChanged*

8.8.9 Configuration attributes

8.8.9.1 Simple attributes

The access terminal shall use as defaults the values in Table 180 typed in ***bold italics***.

Table 180—Configurable simple attributes

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0000	InBandRequestTimer	Soft Allowed/ Static	<i>0x10</i>	The value of the InBandRequestTimer in units of PHY frames.
			0x01–0xff	The value of the InBandRequestTimer in units of PHY frames.
			0x00	Reserved.
0x0001	MinInBandReqRequiredPktSize	Soft Allowed/ Static	<i>0x04</i>	This field in units of times.
			0x01–0x03 and 0x05–0xff	This field in units of eight octets
			0x00	There is no requirement on the access terminal to send InBand Request.
0x0002	MaxRLPacketSize Capability: access network shall not propose this attribute	Soft Allowed/ Static	<i>0x01</i>	The maximum packet size supported by the access terminal on the reverse link, using turbo coding and no extended frames, in units of kilobits, if DuplexingMode = FDD.
			0x02–0xff	The maximum packet size supported by the access terminal on the reverse link, using turbo coding and no extended frames, in units of kilobits, if DuplexingMode = FDD.
			All other values	Reserved
0x0012	MaxRLPacketSizeTDD44 Capability: access network shall not propose this attribute	Soft Allowed/ Static	<i>0x01</i>	The maximum packet size supported by the access terminal on the reverse link using turbo coding and no extended frames, in units of kilobits, if DuplexingMode = TDD44.
			0x02–0xff	The maximum packet size supported by the access terminal on the reverse link using turbo coding and no extended frames, in units of kilobits, if DuplexingMode = TDD44.
			All other values	Reserved.

Table 180—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0022	MaxRLPacketSizeTDD63 Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x01	The maximum packet size supported by the access terminal on the reverse link using turbo coding and no extended frames, in units of kilobits, if DuplexingMode = TDD63.
			0x02–0xff	The maximum packet size supported by the access terminal on the reverse link using turbo coding and no extended frames, in units of kilobits, if DuplexingMode = TDD3.
			All other values	Reserved.
0x0003	MaxRLPacketSizeExtendedFrame Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x01	The maximum packet size for extended frame transmissions supported by the access terminal on the reverse link, using turbo coding and extended frames, in units of kilobits, if DuplexingMode = FDD.
			0x02–0xff	The maximum packet size for extended frame transmissions supported by the access terminal on the reverse link, using turbo coding and extended frames, in units of kilobits, if DuplexingMode = FDD.
			All other values	Reserved.
0x0013	MaxRLPacketSizeExtendedFrameTDD44 Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x01	The maximum packet size for extended frame transmissions supported by the access terminal on the reverse link using turbo coding and extended frames, in units of kilobits, if DuplexingMode = TDD44.
			0x02–0xff	The maximum packet size for extended frame transmissions supported by the access terminal on the reverse link using turbo coding and extended frames, in units of kilobits, if DuplexingMode = TDD44.
			All other values	Reserved.

Table 180—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0023	MaxRLPacketSizeExtendedFrameTDD63 Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x01	The maximum packet size for extended frame transmissions supported by the access terminal on the reverse link using turbo coding and extended frames, in units of kilobits, if DuplexingMode = TDD63.
			0x02–0xff	The maximum packet size for extended frame transmissions supported by the access terminal on the reverse link using turbo coding and extended frames, in units of kilobits, if DuplexingMode = TDD63.
			All other values	Reserved.
0x0004	MaxRLPacketSizeLDPC Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x00	The maximum packet size supported by the access terminal on the reverse link using LDPC coding and no extended frames, in units of kilobits, if DuplexingMode = FDD.
			0x01–0xff	The maximum packet size supported by the access terminal on the reverse link using LDPC coding and no extended frames, in units of kilobits, if DuplexingMode = FDD.
			All other values	Reserved.
0x0014	MaxRLPacketSizeLDPCTDD44 Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x00	The maximum packet size supported by the access terminal on the reverse link using LDPC coding and no extended frames, in units of kilobits, if DuplexingMode = TDD44.
			0x01–0xff	The maximum packet size supported by the access terminal on the reverse link using LDPC coding and no extended frames, in units of kilobits, if DuplexingMode = TDD44.
			All other values	Reserved.

Table 180—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0024	MaxRLPacketSizeLDPCTDD63 Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x00	The maximum packet size supported by the access terminal on the reverse link using LDPC coding and no extended frames, in units of kilobits, if DuplexingMode = TDD63.
			0x01–0xff	The maximum packet size supported by the access terminal on the reverse link using LDPC coding and no extended frames, in units of kilobits, if DuplexingMode = TDD63.
			All other values	Reserved.
0x0005	MaxRLAssignmentSize Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x08	The maximum assignment size that the access terminal can transmit, in units of tiles, if DuplexingMode = FDD.
			0x09–0xff	The maximum assignment size that the access terminal can transmit, in units of tiles, if DuplexingMode = FDD.
			All other values	Reserved
0x0015	MaxRLAssignmentSizeTDD44 Capability: access network shall not propose this attribute		0x08	The maximum assignment size that the access terminal can transmit for non-extended frame assignments, in units of tiles, if DuplexingMode = TDD44.
			0x09–0xff	The maximum assignment size that the access terminal can transmit for non-extended frame assignments, in units of tiles, if DuplexingMode = TDD44.
			All other values	Reserved.
0x0025	MaxRLAssignmentSizeTDD63 Capability: access network shall not propose this attribute		0x08	The maximum assignment size that the access terminal can transmit for non-extended frame assignments, in units of tiles, if DuplexingMode = TDD63.
			0x09–0xff	The maximum assignment size that the access terminal can transmit for non-extended frame assignments, in units of tiles, if DuplexingMode = TDD63.
			All other values	Reserved.

Table 180—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0006	MaxRLAssignmentSizeExtendedFrame Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x08	The maximum assignment size that the access terminal can transmit for extended frame assignments, in units of tiles, if DuplexingMode = FDD.
			0x09–0xff	The maximum assignment size that the access terminal can transmit for extended assignments, in units of tiles, if DuplexingMode = FDD.
			All other values	Reserved.
0x0016	MaxRLAssignmentSizeExtendedFrameTDD44 Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x08	The maximum assignment size that the access terminal can transmit for extended frame assignments, in units of tiles, if DuplexingMode = TDD44.
			0x09–0xff	The maximum assignment size that the access terminal can transmit for extended assignments, in units of tiles, if DuplexingMode = TDD44.
			All other values	Reserved.
0x0026	MaxRLAssignmentSizeExtendedFrameTDD63 Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x08	The maximum assignment size that the access terminal can transmit for extended frame assignments, in units of tiles, if DuplexingMode = TDD63.
			0x09–0xff	The maximum assignment size that the access terminal can transmit for extended assignments, in units of tiles, if DuplexingMode = TDD63.
			All other values	Reserved.
0x0007	MaxCDCHSubsegmentsPerInterface Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x00	The maximum number of CDCH subsegments that the access terminal can transmit on in a given PHY frame.
			0x01–0x20	The maximum number of CDCH subsegments that the access terminal can transmit on in a given PHY frame.
			0x21 to 0xff	Reserved.

Table 180—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0008	MaxNumRLSDMASubtrees Capability: access network shall not propose this attribute	Soft Allowed/ Static	0x02	The maximum number of SDMA subtrees supported by the access terminal on RL.
			0x03–0x08	The maximum number of SDMA subtrees supported by the access terminal on RL.
			All other values	Reserved.
0x0009	MaxOSIMonitorSetSize Capability: access network shall not propose this attribute	Soft Allowed/ Dynamic	0x04	The maximum value of OSIMonitorSetSize supported by the access terminal
			0x05–0x15	The maximum value of OSIMonitorSetSize supported by the access terminal.
			All other values	Reserved.
0x01NN	AttemptBoost AllowedAttributeNN The access terminal shall support all the allowed values of this attribute.	Soft Allowed/ Dynamic	0x00	The value of this field.
			0x00–0x01	The value of this field.
			All other values	Reserved.
0x02NN	MaxCtoIReductionAttributeNN	Soft Allowed/ Dynamic	0x00	The value of this field.
			0x00–0x0f	The value of this field, in units of 1 dB.
			All other values	Reserved.
0x03NN	MaxDeltaIncreaseAttributeNN	Soft Allowed/ Dynamic	0x00	The value of this field.
			0x00–0x03	The value of this field, in units of 1 dB
			All other values	Reserved.
0x04NN	CDMATransmissionIsAllowedAttributeNN	Soft Allowed/ Dynamic	0x00	The value of this field.
			0x00–0x01	The value of this field.
			All other values	Reserved.
0x05NN	CDMATransmissionResondsToRABAttributeNN	Soft Allowed/ Dynamic	0x01	The value of this field.
			0x00–0x01	The value of this field.
			All other values	Reserved.

Table 180—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x06 <i>NN</i>	TerminationTargetAttribute <i>NN</i>	Soft Allowed/ Dynamic	0x03	The value of this field.
			0x00–0x05	The value of this field.
			All other values	Reserved.
0x07 <i>NN</i>	AuxPICHStepDownAttribute <i>NN</i>	Soft Allowed/ Dynamic	0x02	The value of this field.
			0x00–0xff	The value of this field, in units of 0.01 dB.
0x08 <i>NN</i>	AuxPICHStepUpAttribute <i>NN</i>	Soft Allowed/ Dynamic	0x12	The value of this field.
			0x00–0xff	The value of this field, in units of 0.01 dB.

8.8.9.2 Complex attributes

Complex attributes and default values are defined in 8.8.9.2.1 through 8.8.9.2.3 (see 15.2.4 for attribute record definition).

8.8.9.2.1 PADiffBackoffVector

This is a capability attribute, and the access network shall not propose values for this attribute.

This attribute has AttributeID 0x8000. (See Table 181.)

Table 181—Definition of PADiffBackoffVector

Field	Length (bits)	Default
PADiffBackoff1	8	0
PADiffBackoff2	8	0
PADiffBackoff3	8	0

PADiffBackoff1 This entry denotes the maximum additional power in multiples of 0.5 dB (w.r.t. the periodically reported PA Headroom) that the access terminal can output with an assignment that spans less than or equal to 128 contiguous tones.

PADiffBackoff2 This entry denotes the maximum additional power in multiples of 0.5 dB (w.r.t. the periodically reported PA Headroom) that the access terminal can output with an assignment that is localized within a group of 128 contiguous tones that is at least 128 tones away from the edge of the spectral allocation.

PADiffBackoff3 This entry denotes the maximum additional power in multiples of 0.5 dB (w.r.t. the periodically reported PA Headroom) that the access terminal can output with

an assignment consists of 16 contiguous tones (anywhere in the spectral allocation)

Commit	Soft allowed	Scope	Static

8.8.9.2.2 RLMultipleHypothesisDecodeProfileK

This attribute allows the access terminal to send, and the access network to receive more than one packet format for an assignment that contains a given packet format.

This attribute has AttributeID 0x810K. (See Table 182.)

Table 182—Definition of RLMultipleHypothesisDecodeProfileK

Field	Length (bits)	Default
RLMultiDecodeAssignedPFK	4	0
RLMultiDecodeAssignmentSizeK	16	0
RLMultiDecodeNumCompatiblePFK	4	0
RLMultiDecodeNumCompatiblePFK occurrences of the following field		
RLMultiDecodeCompatiblePFKj	4	N/A
Padding	Variable	N/A

RLMultiDecodeAssignedPFK
For the multiple hypothesis profile *K*, this field shall determine the packet format for which the access terminal may send multiple packet formats.

RLMultiDecodeAssignmentSizeK
For the multiple hypothesis profile *K*, this field shall determine the assignment size, in units of base-nodes, for which the access terminal may send multiple packet formats.

RLMultiDecodeNumCompatiblePFK
For the multiple hypothesis profile *K*, this field shall be set to the number of packet formats, in addition to RLMultiDecodeAssignedPF, that the access terminal may transmit for assignment requiring access network multiple decode hypotheses.

RLMultiDecodeCompatiblePFKj
For the multiple hypothesis profile *K*, the *j*th occurrence of this field shall be set to a packet format that the access terminal may send.

Padding
The sender shall set the size of this field to octet align the attribute. This field shall be set to zero by the sender. The receiver shall ignore this field.

Commit	Soft allowed	Scope	Static

8.8.9.2.3 PowerParameters attribute

This attribute has AttributeID 0x8001. (See Table 183.)

Table 183—Definition of PowerParameters

Field	Length (bits)	Default value
DefaultEffectiveTransmitPower	8	23
FastOSIMonitorThreshold	8	40
FastOSIMonitorSetSize	4	2
FastOSISStepUp	8	32
FastOSISStepDown	8	32
OSIMonitorThreshold	8	17
OSIMonitorSetSize	4	2
ChanDiffMax	8	705
ChanDiffMin	8	45
UpDecisionThresholdMin	8	3
DownDecisionThresholdMin	8	7
UpDecisionValue	8	160
DownDecisionValue	8	80
OSI2SequenceMax	8	4
OSISStepUp	8	8
OSISStepDown	8	8
DwUpThreshold	8	136
DwDownThreshold	8	136
DataCoTMin	8	60
DataCoTMax	8	124
AttemptBoostAttribute j , $j = 1, \dots, 6$	4	0
RABMonitorThreshold	8	40
AuxPICHOffsetAttribute	3	3
CDMATrafficPilotOffsetAttribute p , $p = 0, 1$	2	1
InBandPAHTimeout	4	1
MinDataCtoIChange	3	2
MinDeltaSlowChange	3	1
EraseTargetCtoI j , $j = 0, 1, 2, 4$	4	4, 1, 10

Length

Length of the complex attribute in octets. The access network shall set this field to the length of the complex attribute excluding the Length field.

AttributeID	This field shall be set to 0x8001.
ValueID	The sender shall set this field to an identifier assigned to this complex value.
DefaultEffectiveTransmitPower	This field specifies a default value for the EffectiveTransmitPower parameter, to be used to compute ChanDiff for sectors from which EffectiveTransmitPower is not available. The value of this parameter is $20 + n$ dBm.
FastOSIMonitorThreshold	This field specifies a threshold such that only active set members with a ChanDiff value smaller than this threshold shall be added to the FastOSIMonitorSet when the FastOSIEnabled bit is set to '1'. The value of this parameter is $n \times 0.125$ dB.
FastOSIMonitorSetSize	This field specifies the maximum size of the FastOSIMonitorSet.
FastOSIStepUp	This field specifies the up step size for Δ_{tx} adjustments based on fast OSI indications. The value of this parameter is $n \times 2^{-5}$ dB.
FastOSIStepDown	This field specifies the down step size for Δ_{tx} adjustments based on fast OSI indications. The value of this parameter is $n \times 2^{-5}$ dB.
OSIMonitorThreshold	This field specifies a threshold such that only sectors with pilot signal to interference ratios above the threshold shall be added to the OSIMonitorSet. The value of this parameter is $-n$ dB.
OSIMonitorSetSize	This field specifies the maximum size of the OSIMonitorSet. The access network shall not set this parameter to a value larger than the capability parameter MaxOSIMonitorSetSize.
ChanDiffMax	This field determines maximum value of ChanDiff used in determining Δ_{slow} . The value of this parameter is $n - 50$ dB.
ChanDiffMin	This field is determines the minimum value of ChanDiff used in determining Δ_{slow} . The value of this parameter is $n - 50$ dB.
UpDecisionThresholdMin	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this field is $\min(1, 2^{-7} \times n)$.
DownDecisionThresholdMin	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $\min(1, 2^{-7} \times n)$.
UpDecisionValue	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $(n - 128) \times 2^{-5}$.
DownDecisionValue	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $(n - 128) \times 2^{-5}$.
OSI2SequenceMax	This field is set to control parameters internal to the RTC MAC power control algorithm.

OSISStepUp	This field specifies the up step size for Δ_{slow} adjustments based on OSI indications. The value of this parameter is $n \times 2^5$ dB.
OSISStepDown	This field specifies the down step size for Δ_{slow} adjustments based on OSI indications. The value of this parameter is $n \times 2^5$ dB.
DwUpThreshold	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $(n - 128) \times 2^{-5}$ dB.
DwDownThreshold	This field is used to control parameters internal to the RTC MAC power control algorithm. The value of this parameter is $(n - 128) \times 2^{-5}$ dB.
DataCoTMin	This parameter the lower bound for Δ_{slow} adjustments based on OSI indications, and shall take the value $-15 + 0.25 \times n$ dB.
DataCoTMax	This parameter the upper bound for Δ_{slow} adjustments based on OSI indications, and shall take the value $-15 + 0.25 \times n$ dB.
AttemptBoostAttribute _j	The j^{th} instance of this field specifies a boost to be used on the j^{th} HARQ transmission of a RL packet. This parameter shall be in units of dB.
RABMonitorThreshold	This field determines a threshold such that only active set members with a ChanDiff value smaller than this threshold shall be added to the RABMonitorSet. The value of this parameter is $n \times 0.125$ dB.
AuxPICHOffsetAttribute	This field shall be determine the offset of R-AuxPICH with respect to R-PICH. This parameter shall take the value $4 + n$.
CDMATrafficPilotOffsetAttribute _p	This field shall take the value $3 + 0.5n$ dB for $p = 0$, and shall take the value $6 + 0.5n$ for $p = 1$.
InBandPAHTimeout	This parameter specifies the minimum amount of time the access terminal has to wait for an InBand PA headroom report opportunity before being allowed to transmit the report on r-pahch of the R-CDCCH. The value of this parameter is $8 \times n$ PHY frames, where n is the value of the four-bit field.
MinDataCtoIChange	This parameter specifies the minimum change in the DataCtoI value after which the access terminal should include an InBandDataCtoIBlock in the MAC trailer, and shall be in units of dB.
MinDeltaSlowChange	This parameter specifies the minimum change in the Δ_{slow} value after which the access terminal should include an InBandDeltaSlowBlock in the MAC trailer, and shall be in units of dB.
EraseTargetCtoI _j	These parameters specify target C/I values for erasure sequence for different assignment sizes, and shall take the value $-12 + n$ dB, where n is the value of the four-bit field for each parameter.

Commit	Soft allowed	Scope	Static

8.8.10 Protocol numeric constants

The protocol numeric constants are listed in Table 184.

Table 184—Numeric constants of the Reverse Traffic Channel MAC Protocol

Constant	Meaning	Value
$N_{\text{RTCMPTType}}$	Type field for this protocol	See 16.1
$N_{\text{RTCMPBasic}}$	Subtype field for this protocol	0x00
$\text{MinHopportsPerNode}$	Minimum number of hop-ports in a channel tree node	16
$T_{\text{RTCSupervision}}$	Supervision timer	2 s
$N_{\text{RTCMultiHypothesis}}$	Maximum number of MultipleHyothesisDecodeProfiles that may be supported by the access network	16

8.8.11 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

9. Physical Layer

9.1 Basic Physical Layer Protocol

9.1.1 Introduction

9.1.1.1 General overview

An overview of the Basic Physical Layer Protocol is given in 5.4.2.2.4.

9.1.1.2 Primitives and public data

9.1.1.2.1 Commands

This protocol does not define any commands.

9.1.1.2.2 Return indications

This protocol does not return any indications.

9.1.1.2.3 Public data

Subtype for this protocol.

9.1.2 Protocol initialization

9.1.2.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

9.1.2.2 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.2.

9.1.3 Procedures and messages for the InConfiguration instance of the protocol

9.1.3.1 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

9.1.3.2 Message formats

This protocol does not define any messages.

9.1.4 Procedures and messages for the InUse instance of the protocol

9.1.4.1 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

9.1.4.2 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

9.1.4.3 Main procedures

The requirements for the access terminal are described in 9.3.

The requirements for the access network are described in 9.4.

The requirements for Broadcast and Multicast Services, as well as supercast operation are described in 9.5.

9.1.4.4 Interface to other protocols

9.1.4.4.1 Commands

This protocol does not issue any commands.

9.1.4.4.2 Indications

This protocol does not register to receive any indications.

9.1.5 Configuration attributes

This protocol does not define any configuration attributes.

9.1.6 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of only the configuration attributes of this protocol.

9.2 General

9.2.1 Numeric Information

Table 185 lists several variables that are used in the text, including all variables that are defined outside this protocol. In this table, OMP refers to the Overhead Messages Protocol, ECI refers to the ExtendedChannelInfo message, and QCI refers to the QuickChannelInfo block.

Table 185—Physical Layer numeric constants and parameters

Symbol	Meaning	Source	Comments
IsSynchronous	This field determines whether the sector time-base reference is aligned to UTC	IsSynchronous	OMP (AcqInfo)
FDDHalfDuplexTDDPartition	This field determines whether the sector supports half-duplex terminals	FDDHalfDuplexTDDPartition	OMP (AcqInfo)

Table 185—Physical Layer numeric constants and parameters (continued)

Symbol	Meaning	Source	Comments
N_{FFT}	Number of Subcarriers in an OFDM Symbol. Takes values 128, 256, 512, 1024, or 2048	TotalNumSubcarriers	OMP (SystemInfo)
$N_{\text{FFT},B}$	Number of Subcarriers in a BCMCS OFDM Symbol. Takes values 128 or 320	Derived Variable	—
N_{GUARD}	Total number of guard subcarriers	NumGuardSubcarriers	OMP (SystemInfo)
$N_{\text{GUARD},B}$	Total number of guard subcarriers for BCMCS operation	NumGuardSubcarriers	OMP (BroadcastParametersMessage)
T_{CHIP}	Chip duration	Derived Variable	$128/(1.2288 \times N_{\text{FFT}}) \mu\text{s}$
N_{CP}	A multiplicative factor that determines the cyclic prefix duration, where the Cyclic Prefix Duration is $N_{\text{CP}}N_{\text{FFT}}T_{\text{CHIP}}/16$	1, 2, 3, or 4	Determined autonomously by the access terminal using the TDM1 waveform
T_{CP}	Cyclic prefix duration.	Derived Variable	—
$T_{\text{CP},B}$	Cyclic prefix duration for BCMCS	Derived Variable	$T_{\text{CP},B}$
T_s	OFDM symbol duration	Derived Variable	—
$T_{s,B}$	OFDM symbol duration for BCMCS	Derived Variable	—
$T_{\text{SUPERFRAME}}$	Superframe duration	Derived Variable	—
PilotPN	Integer Identifier of the Sector	0–511	—
NumEffectiveAntennas	Number of effective antennas	NumEffectiveAntennas	OMP (QCI)
$Q_{\text{SDMA},\text{RL}}$	Number of SDMA dimensions on the Reverse Link	RLNumSDMADimensions	OMP (ECI)
NumCDMASubSegments _j	Number of CDMA SubSegments. Each entry in the vector denotes the number of CDMA SubSegments on one-eighth of the Reverse Link PHY frames.	NumCDMASubSegments _j	OMP (ECI)
$N_{\text{SUBZONE\_MAX},\text{RL}}$	Size of a subzone on the Reverse Link	RLSubzoneSize	OMP (ECI)
SilenceIntervalPeriod	This field determines the silence interval	SilenceIntervalPeriod	OMP (ECI)
SilenceIntervalDuration	This field determines the silence interval	SilenceIntervalDuration	OMP (ECI)
SilenceIntervalFrequencyMask	This field determines the silence interval	SilenceIntervalFrequencyMask	OMP (ECI)

Table 185—Physical Layer numeric constants and parameters (continued)

Symbol	Meaning	Source	Comments
$N_{\text{SUBZONE\_MAX,FL}}$	Size of a subzone on the Forward Link	FLSubzoneSize	OMP (QCI)
$N_{\text{DRCH-SUBZONES}}$	Number of DRCH Subzones	NumDRCHSubzones	OMP (QCI)
$Q_{\text{SDMA,FL}}$	Number of SDMA dimensions on the Forward Link	FLSDMANumSDMASubtrees	OMP (QCI)
ResourceChannelMux Mode	Resource channel multiplexing mode 1 or 2	ResourceChannelMuxMode	OMP (SystemInfo)
UseDRCHForFLCS	Use DRCH for FLCS	UseDRCHForFLCS	OMP (QCI)
$N_{\text{FLCS-COMMON-BLOCKS}}$	Number of blocks in the FLCS common segment	NumCommonSegmentBlocks	OMP (QCI)
$N_{\text{FLCS-LAB-SEGMENTS}}$	Number of LAB Segments	NumLABSegments	OMP (QCI)
$N_{\text{FLCS-BLOCKS}}$	Number of FLCS hop-ports	Derived Variable	—
$R_{\text{SCCH-BEGIN}}$	Minimum resource index to be used for F-SCCH packets	MinSCCHResourceIndex	OMP (QCI)
$N_{\text{SCCH-MODULATIONSYMBOLS}}$	Number of modulation symbols used by an F-SCCH packet transmitted using QPSK modulation	NumSCCHModulationSymbols	OMP (QCI)
$N_{\text{SCCH-CS}}$	Number of F-SCCH packets in the Common Segment, assuming only QPSK modulation is used	Derived Variable	—
$N_{\text{SCCH-LAB}}$	Number of F-SCCH packets in each LAB segment, assuming only QPSK modulation is used	Derived Variable	—
MaxNumQPSKLABs	Total number of F-SCCH packets, assuming only QPSK modulation is used	Derived Variable	—
R_{T2P}	Traffic-to-pilot ratio to be used for F-SCCH packets transmitted using 16-QAM modulation	16QAMSCCHT2PRatio	OMP (QCI)
$\text{RLDPICHCodeOffset}_{\text{SubtreeIndex}}$	Code offset to be used for the R-DPICH for the subtree with index SubtreeIndex	RLDPICHCodeOffsetSubtreeIndexj	OMP (QCI)
PDCABResourceSharingEnabled	PDCAB Resource Sharing Enabled	PDCABResourceSharingEnabled	OMP (ECI)
SFNID	SFNID of the sector	SFNCellID	OMP (ECI)
EnablePreambleFrequencyReuse	Enable frequency reuse on the superframe preamble	EnablePreambleFrequencyReuse	OMP (AcqInfo)
EnableExpandedQPCH	Enable the transmission of multiple QPCH packets in the same superframe preamble	EnableExpandedQPCH	OMP (QCI)
CPICHHoppingMode	Hopping mode of the CPICH	CPICHHoppingMode	OMP (QCI)

Table 185—Physical Layer numeric constants and parameters (continued)

Symbol	Meaning	Source	Comments
CommonPilotTransmitPower	Takes values “random” or “deterministic”	CommonPilotTransmitPower	OMP (ECI)
FLDPICHCodeOffsetSubtreeIndex		FLDPICHCodeOffsetSubtreeIndex	OMP (QCI)
SinglePAForMultipleCarriers	Determines the modulation of F-BPICH	SinglePAForMultipleCarriers	OMP (QCI)
ShortChannelID	Determines the ChannelBand where the Acquisition Pilot corresponding to an F-BPICH is present.	ShortChannelID	OMP (Sector Parameters)
$N_{\text{CDMA-SUBSEGMENT}}$	The number of subcarriers in a CDMA subsegment with a value of 128	9.2.8.3.2	PHY Constant
MaxPHYSubPacketSize	The maximum subpacket size of a Physical Layer packet	9.2.5	PHY Constant
$N_{\text{CRC,Data}}$	CRC used for the Forward Data Channel, the Reverse CDMA Data Channel, the Reverse OFDMA Data Channel, the Forward Broadcast and Multicast Services Channel, and the Forward Superposed Data Channel	9.2.6.2	PHY Constant
$N_{\text{CRC,PBCCCH}}$	CRC used for the Forward Primary Broadcast Control Channel	9.2.6.2	PHY Constant
$N_{\text{CRC,SBCCH}}$	CRC used for the Forward Secondary Broadcast Control Channel	9.2.6.2	PHY Constant
$N_{\text{CRC,QPCH}}$	CRC used for Forward Quick Paging Channel	9.2.6.2	PHY Constant
$N_{\text{CRC,SCCH}}$	CRC used for the Forward Shared Control Channel	9.2.6.2	PHY Constant
$N_{\text{CRC,ODCCH}}$	CRC used for Forward OFDMA Dedicated Control Channel	9.2.6.2	PHY Constant

9.2.2 System Time

All transmissions from the access network are referenced by System Time, which is a measure of the seconds that have elapsed since the start. Global System Time refers to a common system-wide reference that uses the Global Positioning System (GPS) time scale, which is traceable to, and synchronous with, Coordinated Universal Time (UTC). GPS and UTC differ by an integer number of seconds, specifically the number of leap second corrections added to UTC since January 6, 1980. The start of System Time is January 6, 1980 00:00:00 UTC, which coincides with the start of GPS time.

System Time keeps track of leap second corrections to UTC, but does not use these corrections for physical adjustments to the System Time clocks for global synchronous networks.

A local System Time refers to a time reference that need not be traceable to or synchronous with a common timing reference.

The System Time at various points in the transmission and reception processes is the absolute time referenced at the access network antenna offset by the one-way or round-trip delay of the transmission, as appropriate. (See 3GPP2 C.S0010-C v2.0 and 3GPP2 C.S0011-C v2.0.)

Both Forward Link and Reverse Link transmissions are divided into units of superframes as described in 9.2.7. Each superframe has a SuperframeIndex, which is incremented every superframe. The superframe index t is related to System Time s via the equation $t = \lfloor s/T_{\text{SUPERFRAME}} \rfloor$

9.2.2.1 Synchronization modes

Each sector can be either global synchronous or asynchronous, as determined by the IsSynchronous variable, which is a field of the Overhead Messages Protocol (see 11.6).

In addition, there is the notion of two sectors being mutually synchronous. Information about which sectors are mutually synchronous is carried by the Overhead Messages Protocol (see 11.6) or the Active Set Management Protocol (see 11.7). The timing requirements of mutually synchronous sectors can be found in 3GPP2 C.S0010-C v2.0.

9.2.2.1.1 Synchronous mode

A sector is defined to be in global synchronous mode if its transmit time-base reference is given by the System Time, which is described in 9.2.2. Reliable external means should be provided at each sector operating in global synchronous mode to synchronize its time-base reference to Global System Time. Each sector should use a frequency reference of sufficient accuracy to maintain time alignment to System Time. The sector in global synchronous mode shall maintain the accuracy of its time-alignment with respect to System Time as specified in 3GPP2 C.S0010-C v2.0. The sector shall also maintain the stability of its time-base reference, including the timing drift and rate of timing corrections, as specified in 3GPP2 C.S0010-C v2.0.

Many of the operations in this mode (for example, generation of the TDM pilots) are defined as a function of an auxiliary parameter known as PilotPhase, which changes from superframe to superframe. If a sector is in the synchronous mode, its TDM pilots change from superframe to superframe, thus ensuring that interference seen by the pilots also changes from superframe to superframe for synchronous operation. This enables the access terminal to accumulate processing gain across superframes for the TDM pilots.

9.2.2.1.2 Asynchronous mode

A sector is defined to be in the global asynchronous mode if its transmit time-base reference is given by a Local System Time, that is not aligned to, or synchronous with, the Global System Time. A sector in the global asynchronous mode shall maintain the stability of its time-base reference, including the timing drift and rate of timing corrections, as specified in 3GPP2 C.S0010-C v2.0.

Many of the operations in this mode (e.g., the modulation and scrambling of the TDM pilots) are defined as a function of an auxiliary parameter known as the PilotPhase, which remains unchanged from superframe to superframe. Otherwise, two sectors with different time-base references could possibly have the same PilotPhase at the same time, resulting in a collision of their TDM pilots.

9.2.2.2 Sector identifiers

9.2.2.2.1 PilotPN and PilotPhase

Each sector shall have a 9-bit identifier called the PilotPN. The PilotPN is used in the Physical Layer for differentiating between different sectors. Different sectors with different PilotPNs transmit different acquisition pilots (on the Forward Other Sector Interference Channel), which enable access terminals to monitor the signal strengths of the two sectors. Similarly, two sectors with different PilotPNs use different pseudo-random scrambling sequences on the Forward Data Channel. This in turn ensures that the transmission of one sector appears as noise to an access terminal listening to the other sector.

A 9-bit quantity called PilotPhase is also defined for use in modulating the Forward Other Sector Interference Channel. In the synchronous mode, the PilotPhase of an access network depends on the superframe index and is equal to $(\text{PilotPN} + \text{SuperframeIndex}) \bmod 512$. In the asynchronous mode, the PilotPhase shall equal PilotPN.

9.2.2.2.2 SFNCellID and SFNPhase

SFNCellID is a 9-bit quantity and is a field of the Overhead Messages Protocol. In Synchronous mode, SFNPhase depends on the SuperframeIndex and shall equal $(\text{SFNCellID} + \text{SuperframeIndex}) \bmod 512$. In Asynchronous mode, SFNPhase shall equal SFNCellID.

9.2.2.2.3 PilotID and SectorSeed

The PilotID is a 10-bit quantity that is equal to $[a_0 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0]$ with the rightmost bit being the LSB and the leftmost bit being the MSB, where $[p_8 p_7 \dots p_0]$ is the 9-bit PilotPN and the bit a_0 is given by the IsSynchronous field of the Overhead Messages Protocol. Note that the PilotID is useful in distinguishing between two sectors with the same PilotPN provided that they have different values of the IsSynchronous field.

A 20-bit quantity known as the SectorSeed is also defined in each frame in each superframe. The SectorSeed shall be equal to $[a_0 p_8 p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0 s_3 s_2 s_1 s_0 f_5 f_4 f_3 f_2 f_1 f_0]$ with the rightmost bit being the LSB and the leftmost bit being the MSB, where $[a_0 p_8 p_7 \dots p_0]$ is the 10-bit PilotID, $[s_3 \dots s_0]$ are the four LSBs of the superframe index, and $[f_5 f_4 f_3 f_2 f_1 f_0]$ are the 6 LSBs of the frame index (within the superframe) (see Figure 63). For transmissions in the superframe preamble, the 6-bits $[f_5 \dots f_0]$ shall be set to $[111111]$. a_0 is provided by the IsSynchronous field the Overhead Messages Protocol.

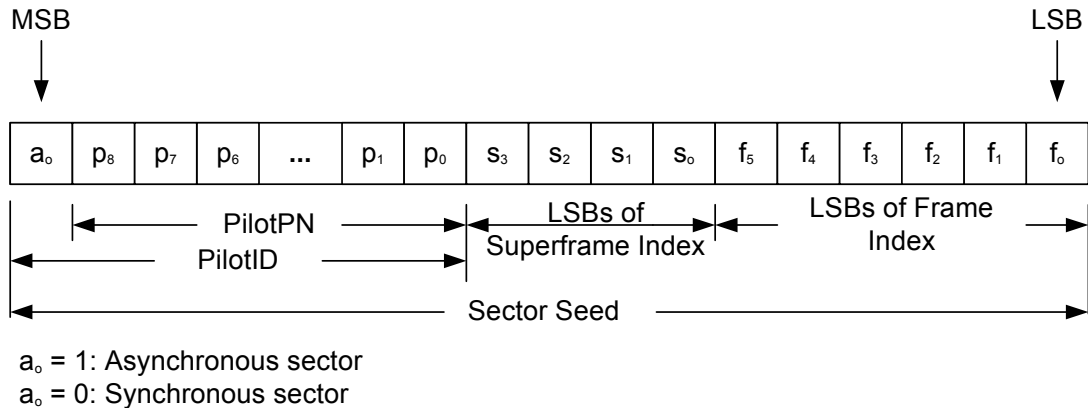


Figure 63—Definition of Sector Seed

The SectorSeed is used in generating scrambling sequences for almost all Forward and Reverse Link channels.

9.2.2.3 Access terminal time-base reference

The access terminal shall establish a time-base reference for each sector to which it transmits. The time-base reference is used to derive timing for various time-critical transmission components, including superframe boundaries, PHY frame boundaries, etc.

The access terminal initial time-base reference for a sector shall be established from the acquired TDM Pilot 1, TDM Pilot 2, and TDM Pilot 3 OFDM symbols, as well as from the SystemTimeLSB field of the Overhead Messages Protocol, where, TDM Pilot 1, TDM Pilot 2, and TDM Pilot 3 are defined in 9.2.7.1.4. The initial access terminal time-base reference shall coincide with the time of occurrence, as measured at the access terminal antenna connector, of the earliest arriving multipath component of the Forward Link waveform. Thus, the beginning of the Reverse Link superframe with index i shall coincide with the beginning of the Forward Link superframe with index i , where the beginning of both superframes are measured at the access terminal antenna connector (see Figure 64).

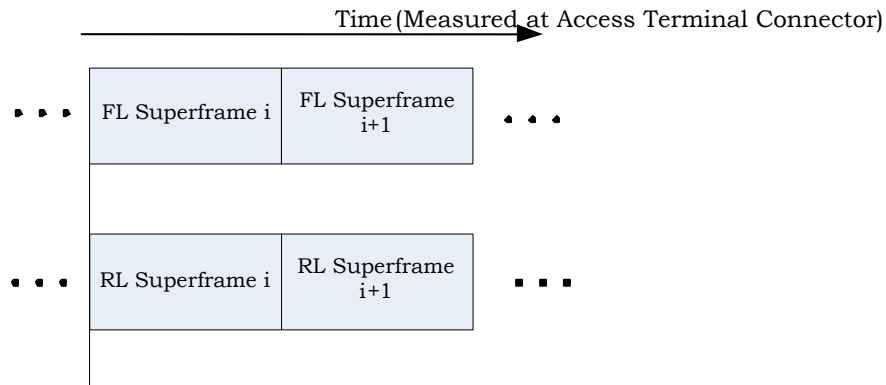


Figure 64—Relationship between the forward link and reverse link timings

After the initial time-base reference has been established, the access terminal shall advance and retard the time-base reference of the Reverse Link Serving Sector in response to the AccessGrant block of the FLCS MAC Protocol (see 8.5.5.4) and the TimingCorrection message of the Connected State Protocol (see 8.5.6.1) from that sector, as specified in the following paragraphs.

To advance timing by a period of k chips, the access terminal shall move its time-base reference earlier by a period of kT_c , where T_c is the chip duration. To retard timing by a period of k chips, the access terminal shall move its time-base reference later by a period of kT_c . The timing correction messages from a given sector are with respect to the Reverse Pilot Channel transmission of the access terminal on the CDMA subsegments assigned by that sector.

The access terminal may use the AccessGrant and TimingCorrection messages from sectors other than the Reverse Link Serving Sector in order to update its time-base references for those sectors. For these sectors, the access terminal may update the time-base reference of a sector using the time-base reference of another sector that is mutually synchronous with it, taking into account the difference in time of occurrence, as measured at the access terminal antenna connector, of the earliest arriving multipath component of the Forward Link waveform from the two sectors. The Active Set Management Protocol (see 11.7) determines whether two sectors are mutually synchronous with each other.

Upon a successful handoff using the CQI/REQ-repointing (see 8.7.5.4), the time reference for the new FLSS/RLSS shall be given by the time-base reference of the CQI/REQ transmission that initiated the handoff. Upon a successful handoff based on access probe transmission (see 8.4.5.5), the time reference for the new serving sector shall be given by the timing in the AccessGrant message.

If the access terminal maintains independent time-base references for transmission to two or more sectors, then it shall generate a time-domain waveform (as described in 9.2.7.3) using the time-base reference for each such sector and shall transmit the sum of the time-domain waveforms over the air.

9.2.2.4 DuplexingMode

Each sector shall operate in one of three duplexing modes: FDD, TDD44, and TDD63. The choice of duplexing mode is determined by the DuplexingMode parameter of the Overhead Messages Protocol. The superframe partitioning is different for different duplexing modes and is described in 8.1.4.

Furthermore, if DuplexingMode = FDD, then the sector may operate either with FDDHalfDuplexTDDPartition = 0 to disable half-duplex mode or with FDDHalfDuplexTDDPartition = 1 to enable half-duplex mode.

9.2.3 Tolerances

Unless otherwise specified, all values indicated are exact unless an explicit tolerance is stated.

9.2.4 Reserved bits

Some bits are marked as reserved bits in the frame structure of some channels. The access terminal and the access network shall ignore the state of all bits that are marked as reserved bits in all frames that they receive.

9.2.5 Common Physical Layer algorithms and definitions

9.2.5.1 Common permutation generation algorithm

The algorithm takes a 20-bit seed and a permutation size M as inputs and outputs a permutation of the set $\{0, 1, \dots, M-1\}$. The algorithm uses a linear feedback shift register to generate pseudorandom numbers, which in turn are used to generate pseudorandom permutations.

The 20-tap linear feedback shift register shall have a generator sequence of $h(D) = 1 + D^{17} + D^{20}$, as shown in Figure 65. The j^{th} output $I(j)$ of this shift register shall satisfy $I(j) = I(j-17) \oplus I(j-20)$.

The initial state of the register shall generate the first output bit. A pseudorandom number x in $\{0, 1, \dots, 2^n-1\}$ for any $n < 20$ can be generated by clocking the register n times, with the initial output bit being the LSB of x and the final (n^{th}) output bit being the MSB of x .

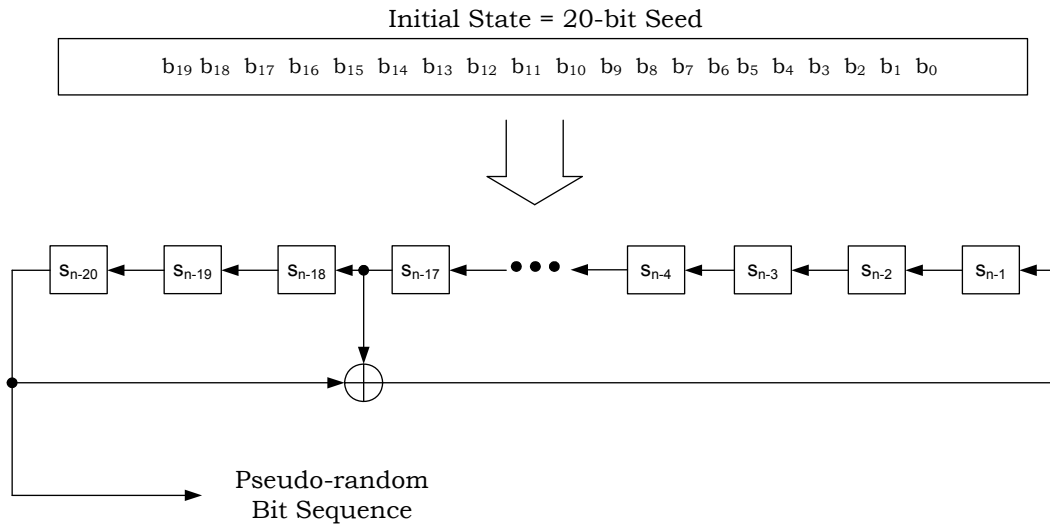


Figure 65—PN Register for generating pseudorandom bits

The common permutation generation algorithm shall generate a permutation of size M as follows:

- a) Initialization steps:
 - 1) Let n be the smallest integer such that $M \leq 2^n$.
 - 2) Initialize an array A of size M such that $A[0] = 0$, $A[1] = 1$..., $A[M - 1] = M - 1$.
 - 3) Initialize the PN register with the 20-bit seed.
 - 4) Initialize counter i to $M - 1$.
- b) Repeat the following steps until $i = 0$.
 - 1) Find the smallest p such that $i < 2^p$.
 - 2) Initialize a counter j to 0. Initialize x to $(i + 1)$. Repeat the following steps until $x = i$ or until $j = 3$.
 - i) Clock the PN register n times to obtain an n -bit pseudorandom number y .
 - ii) Let $x = (y \bmod 2^p)$.
 - iii) Increment j by 1.
 - 4) If $x > i$, set $x = x - i$.
 - 5) Swap the i^{th} and the x^{th} elements in the array (i.e., perform the steps $TMP = A[i]$; $A[i] = A[x]$; $A[x] = TMP$).
 - 6) Decrement i .

The output permutation P is generated from the resulting array A as follows: $P(y) = A(M - 1 - y)$ for all y . For example, if A reads 534021, then $P(0) = 1$, $P(1) = 2$, $P(2) = 0$, $P(3) = 4$, $P(4) = 3$, and $P(5) = 5$.

9.2.5.2 Pruned bit reversal interleaver

The pruned bit reversal interleaver generates a permutation $y = \text{PBRI}(i, M)$ of the sequence of $\{0, 1, \dots, M - 1\}$ of size M , where y is the output value corresponding to the input value i . The pruned bit reversal interleaver is defined as follows:

- 1) Determine the pruned bit-reversal interleaver parameter, n , where n is the smallest integer such that $M \leq 2^n$.
- 2) Initialize counters i and j to 0.
- 3) Define x as the bit-reversed value of j using an n -bit binary representation. For example, if $n = 4$ and $j = 3$, then $x = 12$.
- 4) If $x < M$, set $\text{PBRI}(i, M)$ to x and increment the counter i by 1.
- 5) Increment the counter j by 1.
- 6) If $(i < M)$ go to 3.

9.2.5.3 Common real and complex scrambling algorithms

The common real and complex scrambling algorithms use a 20-bit seed as input and output a sequence of real and complex scrambling symbols respectively.

For both algorithms, the input seed shall be denoted as a 20-bit number $S_{\text{INPUT}} = [b_{19} \ b_{18} \dots \ b_1 \ b_0]$. If the input seed has less than 20-bits, it shall be padded with zeros as MSBs to generate the 20-bit seed S_{INPUT} . If the input seed has more than 20-bits, then the seed S_{INPUT} shall be set to be the 20 MSBs of the input seed.

The i^{th} entry $c(i)$ in the complex scrambling sequence shall be generated from two bits, denoted by $b_0(i)$ and $b_1(i)$, using the mapping described in Table 186. The bits $b_0(i)$ and $b_1(i)$ shall be the $2i^{\text{th}}$ and $(2i+1)^{\text{th}}$ bits in the pseudo-random bit sequence generated as described in 9.2.5.3.1.

Table 186—Generation of complex scrambling symbols

$b_0(i)$	$b_1(i)$	$c(i)$
0	0	(0, 1)
0	1	(1, 0)
1	0	(0, -1)
1	1	(-1, 0)

The i^{th} entry $r(i)$ in the real scrambling sequence shall be generated from a bit denoted by $b_r(i)$, using the mapping $r(i) = (1 - 2 \ b_r(i))$. The bit $b_r(i)$ shall be the i^{th} bit in the pseudo-random bit sequence generated as described in 9.2.5.3.1.

9.2.5.3.1 Pseudo-random bit sequence generation for scrambling

The register shall have a generator polynomial $h_I(D) = D^{20} + D^{19} + D^{16} + D^{14} + 1$ i.e., the j^{th} output $I(j)$ of the register shall satisfy $I(j) = I(j - 20) \oplus I(j - 19) \oplus I(j - 16) \oplus I(j - 14)$. The initial state of the register shall be set to S_{INPUT} and used to generate the initial scrambling bit. The n^{th} scrambling bit shall be generated by clocking the register n times.

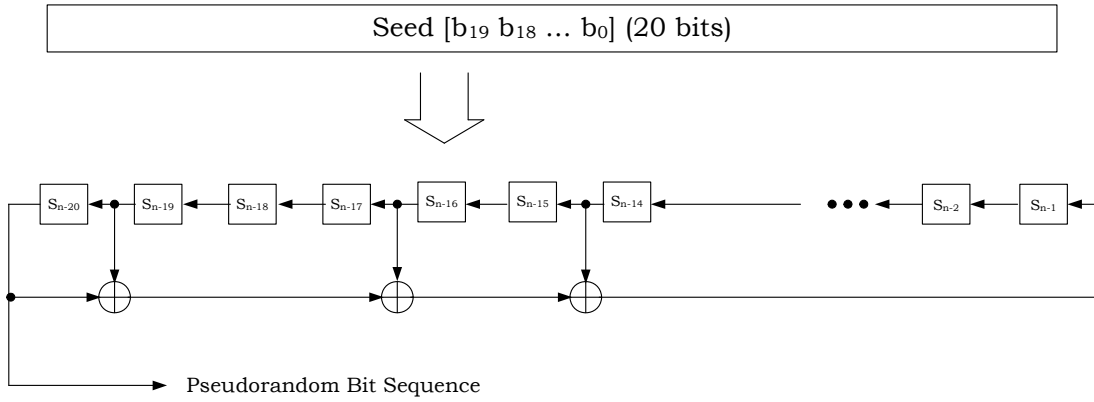


Figure 66—Scrambling sequence register

9.2.5.4 Common PHY hash function

The common PHY hash function takes a non-negative integer $x < 2^{64}$ and returns a 20-bit output. The output y shall be computed as follows:

- Set x_1 to be $x \bmod 2^{32}$ and x_2 to be $\lfloor x/2^{32} \rfloor$.
- Set $\text{TMP}_1 = [x_1 \times 2654435761] \bmod 2^{32}$ and $\text{TMP}_2 = [x_2 \times 2654435761] \bmod 2^{32}$.
- Set y_1 to be the 20 LSBs of the bit-reversed value of TMP_1 in a 32-bit representation, i.e., $y_1 = [\text{Bit-Reverse}(\text{TMP}_1)] \bmod 2^{20}$. Set y_2 to be the 20 LSBs of the bit-reversed value of TMP_2 in a 32-bit representation, i.e., $y_2 = [\text{Bit-Reverse}(\text{TMP}_2)] \bmod 2^{20}$.
- Set y to be the bit-wise XOR of y_1 and y_2 .

The common PHY hash function is denoted as $f_{\text{PHY-HASH}}$ i.e., $y = f_{\text{PHY-HASH}}(x)$.

9.2.5.5 Discrete Fourier Transform (DFT)

The DFT of an length- N sequence X with elements $x_0, x_1, \dots, x_{N-1}$ is given by another length- N sequence Y with elements $y_0, y_1, \dots, y_{N-1}$. The elements of Y are related to the elements of X via the relationship shown in Equation (16).

$$y_f = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} x_t e^{-j2\pi(f - \frac{N}{2})t/N} \quad (16)$$

The DFT matrices are defined as $N \times N$ matrices D , where D is defined as shown in Equation (17).

$$D = [d_{nm}] = \left[\frac{1}{\sqrt{N}} e^{-j2\pi(n - \frac{N}{2})m/N} \right] \quad (17)$$

9.2.5.6 Walsh sequence

A Walsh sequence W_i^N , where N is a power of 2 and i is a non-negative integer less than N , is a length N binary sequence taking on $\{-1, 1\}$, which is given by the i^{th} column of the $N \times N$ Hadamard matrix W^N . The $N \times N$ Hadamard matrix W^N is conventionally defined by the recursive relationship shown in Equation (18):

$$W^2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, W^{2N} = \begin{bmatrix} W^N & W^N \\ W^N & -W^N \end{bmatrix} \quad (18)$$

The Walsh sequence W_i^N may also be referred to as the Walsh sequence of length N with index i .

9.2.6 Coding and modulation

9.2.6.1 Coding and modulation structures

Coding and modulation structures common to both the Forward and Reverse Links are illustrated in Figure 67. The packet splitting applies to the Reverse CDMA Data Channel, the Reverse OFDMA Data Channel, the Forward Data Channel, the Forward Broadcast and Multicast Services Channel, and the Forward Superposed Data Channel. The input packets shall be converted into one or more subpackets for transmission, and the sequence of CRC insertion, encoding, channel interleaving, sequence repetition, and data scrambling operations shall be performed independently for each subpacket. All channels other than the Reverse CDMA Data Channel, the Reverse OFDMA Data Channel, the Forward Data Channel, the Forward Broadcast and Multicast Services Channel, and the Forward Superposed Channel shall use a single sequence of CRC insertion, encoding, channel interleaving, sequence repetition, and data scrambling operations. For these channels, the term packet and subpacket may be used interchangeably.

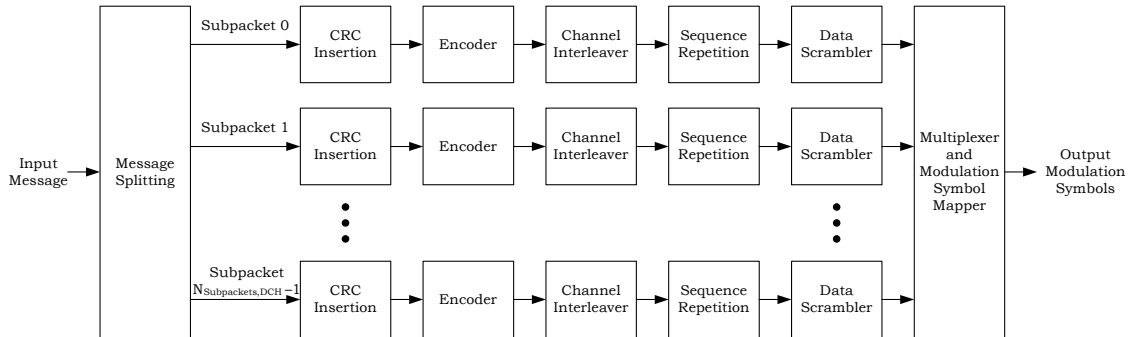


Figure 67—Coding and modulation structure

If the input packet size $N_{\text{PACKET\_BITS}}$ is larger than $\text{MaxPHYSubPacketSize}$, the packet shall be split into $N_{\text{SUBPACKETS}}$ subpackets, indexed from 0 to $N_{\text{SUBPACKETS}} - 1$, where $N_{\text{SUBPACKETS}} = \lceil N_{\text{PACKET\_BITS}} / \text{MaxPHYSubPacketSize} \rceil$ where, $\text{MaxPHYSubPacketSize}$ is equal to a constant of the Physical Layer Protocol. When $N_{\text{PACKET\_BITS}}$ is less than $\text{MaxPHYSubPacketSize}$, there shall only be one subpacket.

Define

$$t_0 = \left(\frac{N_{\text{PACKET\_BITS}}}{8} \right) \bmod N_{\text{SUBPACKETS}}$$

$$t_1 = N_{\text{SUBPACKETS}} - t_0$$

$$b_0 = 8 \left\lceil \frac{N_{\text{PACKET\_BITS}}}{8N_{\text{SUBPACKETS}}} \right\rceil$$

$$b_1 = \begin{cases} b_0, & \text{if } t_0 = 0 \\ b_0 - 8, & \text{otherwise} \end{cases}$$

where $N_{\text{SUBPACKETS}}$ is the number of subpackets in the packet and $N_{\text{PACKET\_BITS}}$ is the number of bits in the packet ($N_{\text{PACKET\_BITS}}$ is a multiple of 8 bits). Then, the first t_0 subpackets shall each have b_0 bits, and the last t_1 subpackets shall each have b_1 bits. The packet bits shall be distributed to the different subpackets in order. That is, the first set of packet bits shall form the first subpacket, the next of packet bits shall form the second subpacket, etc.

At the receiver, a packet shall be declared to be in error if any of the constituent subpackets of the packet are in error.

9.2.6.2 Error detection

Cyclic Redundancy Code (CRC) bits are used to detect errors in the received subpackets for some Forward and Reverse Link channels. The CRC bits are appended to the input information bits.

The number of CRC bits generated for all Forward and Reverse Link channels shall be as specified in Table 187.

Table 187—Number of CRC bits for the Forward and Reverse Link Channels

Channel	Number of CRC bits
Reverse Pilot Channel	None
Reverse Auxiliary Pilot Channel	None
Reverse Access Channel	None
Reverse Code Division Multiple Access Dedicated Control Channel	None
Reverse Code Division Multiple Access Data Channel	24
Reverse Dedicated Pilot Channel	None
Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel	9
Reverse Acknowledgment Channel	None
Reverse Orthogonal Frequency Division Multiple Access Data Channel	24
Forward Preamble Pilot Channel	None
Forward Other Sector Interference Channel	None
Forward Primary Broadcast Control Channel	12

**Table 187—Number of CRC bits for the
Forward and Reverse Link Channels (continued)**

Channel	Number of CRC bits
Forward Secondary Broadcast Control Channel	12
Forward Acquisition Channel	None
Forward Beacon Pilot Channel	None
Forward Null Channel	None
Forward Quick Paging Channel	12
Forward Common Pilot Channel	None
Forward Channel Quality Indicator Pilot Channel	None
Forward Superposed Channel Quality Indicator Pilot Channel	None
Forward Dedicated Pilot Channel	None
Forward Acknowledgment Channel	None
Forward Start of Packet Channel	None
Forward Shared Control Channel (non-GRA Block)	16
Forward Shared Control Channel (GRA Block)	5
Forward Pilot Quality Indicator Channel	None
Forward Fast Other Sector Interference Channel	None
Forward Reverse Activity Bit Channel	None
Forward Interference-Over-Thermal Channel	None
Forward Power Control Channel	None
Forward Data Channel	24
Forward Broadcast and Multicast Services Channel	24
Forward Superposed Data Channel	24

9.2.6.2.1 Generation of the CRC bits

The CRC bits shall be computed according to the following procedure (see Figure 68):

- Initially, all the switches shall be set in the up position and the shift-register elements shall be set to logical one.
- The register shall be clocked a number of times equal to the number of input bits in the subpacket with those bits as input.
- The switches shall be set in the down position so that the output is a modulo-2 addition with a '0' and the successive shift register inputs are '0's.
- The register shall be clocked an additional number of times equal to the number of CRC bits.
- These additional bits shall be the CRC bits.
- The bits shall be transmitted in the order calculated.

The generator polynomial for the 24-bit CRC shall be

$g(x) = x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$. The Cyclic Redundancy Code of length 24 can be generated by the shift-register structure shown in Figure 68.

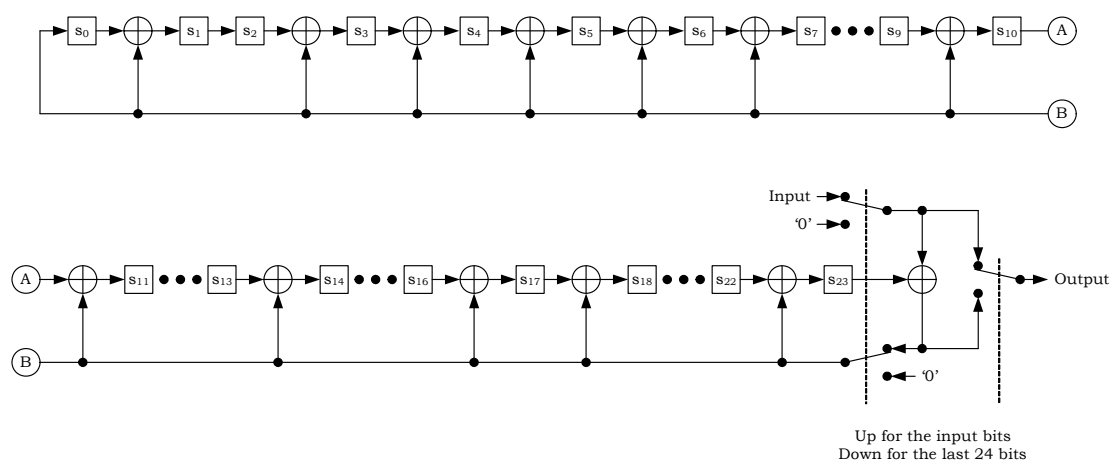


Figure 68—Calculations for the 24-bit CRC

When the CRC length is equal to $N_{CRC} < 24$, 24 CRC bits shall be computed as described above. However, only the first N_{CRC} bits shall be transmitted and the remaining bits shall be discarded.

9.2.6.3 Forward error correction

Table 188 specifies the types of forward error correction that shall be used.

Table 188—Types of forward error correction for the Forward and Reverse Link Channels

Channel	Type of Coding
Reverse Pilot Channel	None
Reverse Auxiliary Pilot Channel	None
Reverse Access Channel	None
Reverse Code Division Multiple Access Dedicated Control Channel	None
Reverse Code Division Multiple Access Data Channel	Rate-1/5 Turbo or Rate-1/3 Convolutional
Reverse Dedicated Pilot Channel	None
Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel	Rate-1/3 Convolutional
Reverse Acknowledgment Channel	None
Reverse Orthogonal Frequency Division Multiple Access Data Channel	Rate-1/5 Turbo, LDPC, or Rate-1/3 Convolutional
Forward Preamble Pilot Channel	None
Forward Primary Broadcast Control Channel	Rate-1/3 Convolutional
Forward Secondary Broadcast Control Channel	Rate-1/3 Convolutional
Forward Acquisition Channel	None

**Table 188—Types of forward error correction for the
Forward and Reverse Link Channels (continued)**

Channel	Type of Coding
Forward Beacon Pilot Channel	None
Forward Null Channel	None
Forward Quick Paging Channel	Rate-1/3 Convolutional
Forward Other-Sector-Interference Channel	None
Forward Common Pilot Channel	None
Forward Channel Quality Indicator Pilot Channel	None
Forward Superposed Channel Quality Indicator Pilot Channel	None
Forward Dedicated Pilot Channel	None
Forward Acknowledgment Channel	None
Forward Start-of-Packet Channel	None
Forward Shared Control Channel	Rate-1/3 Convolutional or Rate-1/3 Tail-biting Convolutional
Forward Pilot Quality Indicator Channel	Rate-1/3 Concatenated
Forward Fast Other-Sector-Interference Channel	Rate-1/3 Concatenated
Forward Fast Interference-over-Thermal Channel	Rate-1/3 Concatenated
Forward Power Control Channel	None
Forward Reverse Activity Bit Channel	None
Forward Data Channel	Rate-1/3 Convolutional, LDPC, or Rate-1/5 Turbo
Forward Broadcast and Multicast Services Channel	Rate-1/5 Turbo
Forward Superposed Data Channel	Rate-1/3 Conventional, LDPC, or Rate-1/5 Turbo

9.2.6.3.1 Convolutional encoding

A rate-1/3 convolutional code shall be used to encode CRC-padded subpackets on the Forward Primary Broadcast Control Channel, the Forward Secondary Broadcast Control Channel, the Forward Shared Control Channel, and the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel. It shall also be used to encode the CRC-padded subpackets of the Forward Data Channel, the Reverse CDMA Data Channel, and the Reverse Orthogonal Frequency Division Multiple Access Data Channel when those CRC-padded subpackets have less than or equal to 128 bits. The input to the encoder shall consist of the bits of the CRC-padded subpacket appended with eight all-zero encoder tail bits.

The generator functions for the rate-1/3 code shall be as follows: g_0 equals 557 (octal), g_1 equals 663 (octal), and g_2 equals 711 (octal). This code generates three encoder output bits for each bit that is input to the encoder. These encoder output bits shall be output so that the bit (c_0) encoded with generator function g_0 is output first, the bit (c_1) encoded with generator function g_1 is output second, and the bit (c_2) encoded with generator function g_2 is output last. The state of the convolutional encoder, upon initialization, shall be the

all-zero state. The first encoder output bit that is output after initialization shall be a bit encoded with generator function g_0 .

Convolutional encoding involves the modulo-2 addition of selected taps of a serially time-delayed data sequence. The length of the data sequence delay is equal to $K - 1$, where K is the constraint length of the code. Figure 69 illustrates the specific $K = 9$, rate-1/3 convolutional encoder that is used.

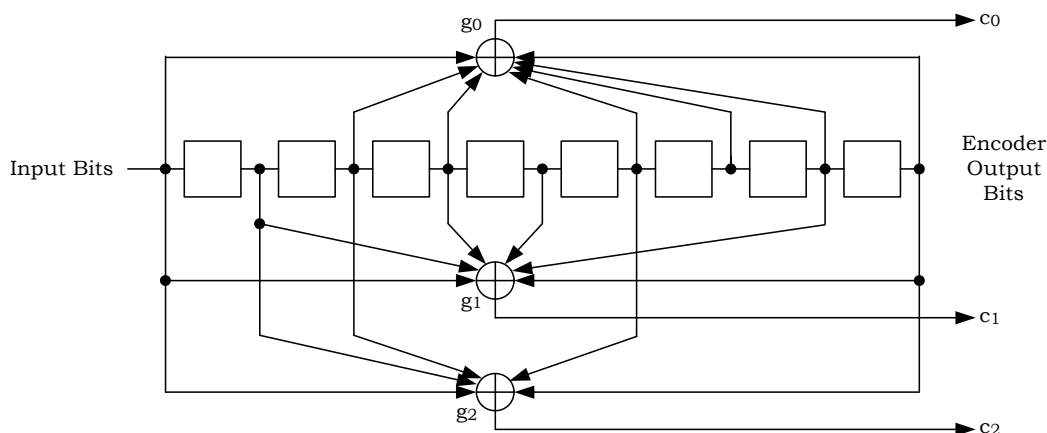


Figure 69— $K = 9$, Rate-1/3 Convolutional Encoder

9.2.6.3.1.1 Rate-1/3 tail-biting convolutional encoding

A rate-1/3 tail-biting convolutional code may be used to encode CRC-padded subpackets on the Forward Shared Control Channel. The rate-1/3 tail-biting convolutional code takes in N input bits and outputs $3N$ output bits according to the following procedure:

- Let $X_0, X_1, \dots, X_{N-1}$ be the sequence of N input bits.
- Let Y be the sequence of $3(N + 8)$ bits obtained by encoding the sequence of $(N + 8)$ bits $(X_{N-8}, X_{N-7}, \dots, X_{N-1}, X_0, X_1, \dots, X_{N-1})$ using the rate-1/3 convolutional code described in 9.2.6.3.1. No additional zero bits are appended to the $(N + 8)$ input bits before the convolutional encoding operation.
- The output of the tail-biting code shall be the last $3N$ bits of Y , i.e., bits $Y_{24}, Y_{25}, \dots, Y_{3N+23}$.

9.2.6.3.2 Rate-1/3 concatenated encoding

The block code as shown in Table 189 is used to encode packets on the Forward Pilot Quality Indicator Channel, Forward Fast Other Sector Interference Channel, and Forward Interference over Thermal Channel. This code may be implemented using a combination of a two-bit Cyclic Code and a rate-1/2 tail-biting convolutional code.

9.2.6.3.2.1 Cyclic code generation

The generator polynomial for the Cyclic Code shall be

$$g(x) = x^2 + 1$$

Two parity bits using the cyclic code shall be generated by the shift-register structure shown in Figure 70.

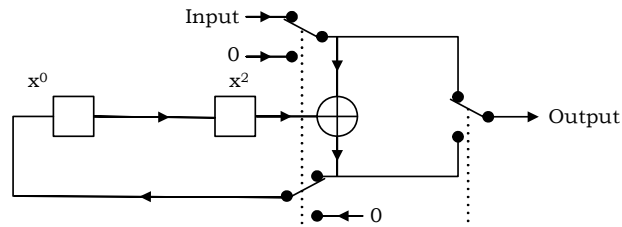


Figure 70—Calculations for the Cyclic Code

The parity bits shall be computed according to the following procedure:

- Initially, all the switches shall be set in the up position and the shift-register elements shall be set to logical zero (0).
- The register shall be clocked a number of times equal to the number of input bits in the subpacket with those bits as input.
- The switches shall be set in the down position so that the output is a modulo-2 addition with a '0' and the successive shift register inputs are '0's.
- The register shall be clocked an additional number of times equal to the number of parity bits.
- These additional bits shall be the parity bits.
- The bits shall be transmitted in the order calculated.

9.2.6.3.2.2 Rate-1/2 tail-biting convolutional code generation

The generator functions for the tail-biting convolutional code shall be as follows: g_0 equals 165 (octal) and g_1 equals 173 (octal). This code generates two code symbols for each data bit input to the encoder. These code symbols shall be output so that the code symbol (c_0) encoded with generator function g_0 shall be output first and the code symbol (c_1) encoded with generator function g_1 shall be output last. The state of the convolutional encoder, upon initialization, shall be the input data bits and shall be the same as that after encoding. The first code symbol output after initialization shall be a code symbol encoded with generator function g_0 as shown in Figure 71.

Convolutional encoding involves the modulo-2 addition of selected taps of a serially time-delayed data sequence. The length of the data sequence delay is equal to $K - 1$, where K is the constraint length of the code. Figure 71 illustrates the specific $K = 7$, rate-1/2 convolutional encoder that is used.

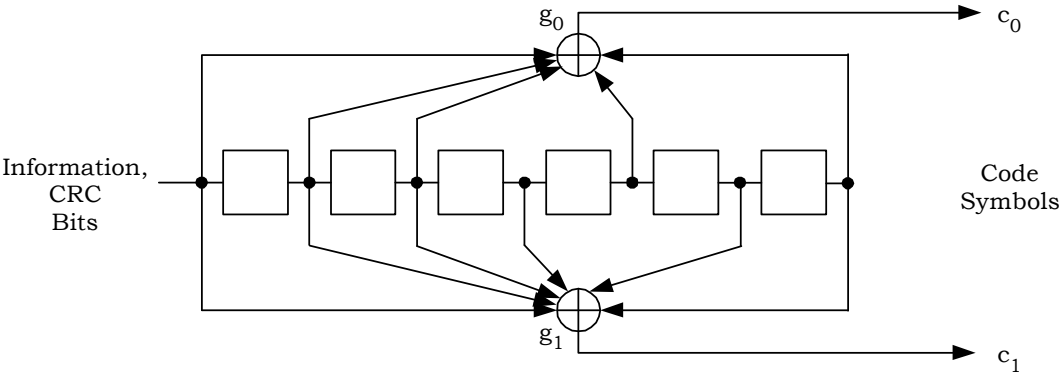


Figure 71—K=7, Rate-1/2 convolutional encoder

9.2.6.3.2.3 Block code description

Equivalently, the cyclic code and convolutional encoding can be represented as a linear block code as shown in Table 189.

Table 189—Codewords for the concatenated code

Input data	Output codewords
0000	0000 0000 0000
0001	1001 0010 1011
0010	0100 1010 1110
0011	1101 1000 0101
0100	1011 1001 0010
0101	0010 1011 1001
0110	1111 0011 1100
0111	0110 0001 0111
1000	1110 0100 1010
1001	0111 0110 0001
1010	1010 1110 0100
1011	0011 1100 1111
1100	0101 1101 1000
1101	1100 1111 0011
1110	0001 0111 0110
1111	1000 0101 1101

9.2.6.3.3 Turbo encoding

A rate-1/5 turbo code shall be used to encode the CRC-padded subpackets of the Forward Data Channel, the Forward Broadcast and Multicast Services Channel, the Forward Superposed Channel, the Reverse CDMA Data Channel, and the Reverse Orthogonal Frequency Division Multiple Access Data Channel when those CRC-padded subpackets have greater than 128 bits. CRC-padded subpackets consist of information bits of a Forward Data Channel, a Forward Broadcast and Multicast Services Channel, a Forward Superposed Channel, or a Reverse CDMA Data Channel subpacket and CRC bits. The input to the encoder shall consist of the bits of the CRC-padded subpacket.

The turbo code is a parallel concatenation of two constituent systematic, recursive, convolutional codes with a turbo interleaver preceding the second recursive convolutional encoder.

9.2.6.3.3.1 Turbo encoder

The transfer function for the constituent codes shall be as shown in Equation (19).

$$G(D) = \begin{bmatrix} 1 & \frac{n_0(D)}{d(D)} & \frac{n_1(D)}{d(D)} \end{bmatrix} \quad (19)$$

where

$$d(D) = 1 + D^2 + D^3$$

$$n_0(D) = 1 + D + D^3$$

$$n_1(D) = 1 + D + D^2 + D^3$$

The turbo encoder shall generate an output bit sequence that is identical to the one generated by the encoder shown in Figure 72. Initially, the states of the constituent encoder registers in this figure are set to zero. Then, the constituent encoders are clocked with the switches in the positions noted.

The turbo encoder generates $5N_{\text{TURBO}} + 18$ encoder output bits, where N_{TURBO} is the number of encoder input bits. The first $5N_{\text{TURBO}}$ encoder output bits shall be generated by clocking the constituent encoders once for each encoder input bit with the switches in the up positions and then puncturing the X' encoder output bits. The sequence of encoder output bits for each encoder input bit shall be $XY_0Y_1Y'_0Y'_1$ with the X bit output first.

The last 18 encoder output bits are called the encoder output tail bits. These tail bits shall be generated after the constituent encoders have been clocked N_{TURBO} times with the switches in the up positions. The first nine encoder output tail bits shall be generated by clocking Constituent Encoder 1 three times with its switches in the down position while Constituent Encoder 2 is not clocked. The sequence of encoder output bits for each clocking of Constituent Encoder 1 shall be XY_0Y_1 . The last nine encoder output tail bits shall be generated by clocking Constituent Encoder 2 three times with its switches in the down position while Constituent Encoder 1 is not clocked. The sequence of encoder output bits for each clocking of Constituent Encoder 2 shall be $X'Y'_0Y'_1$.

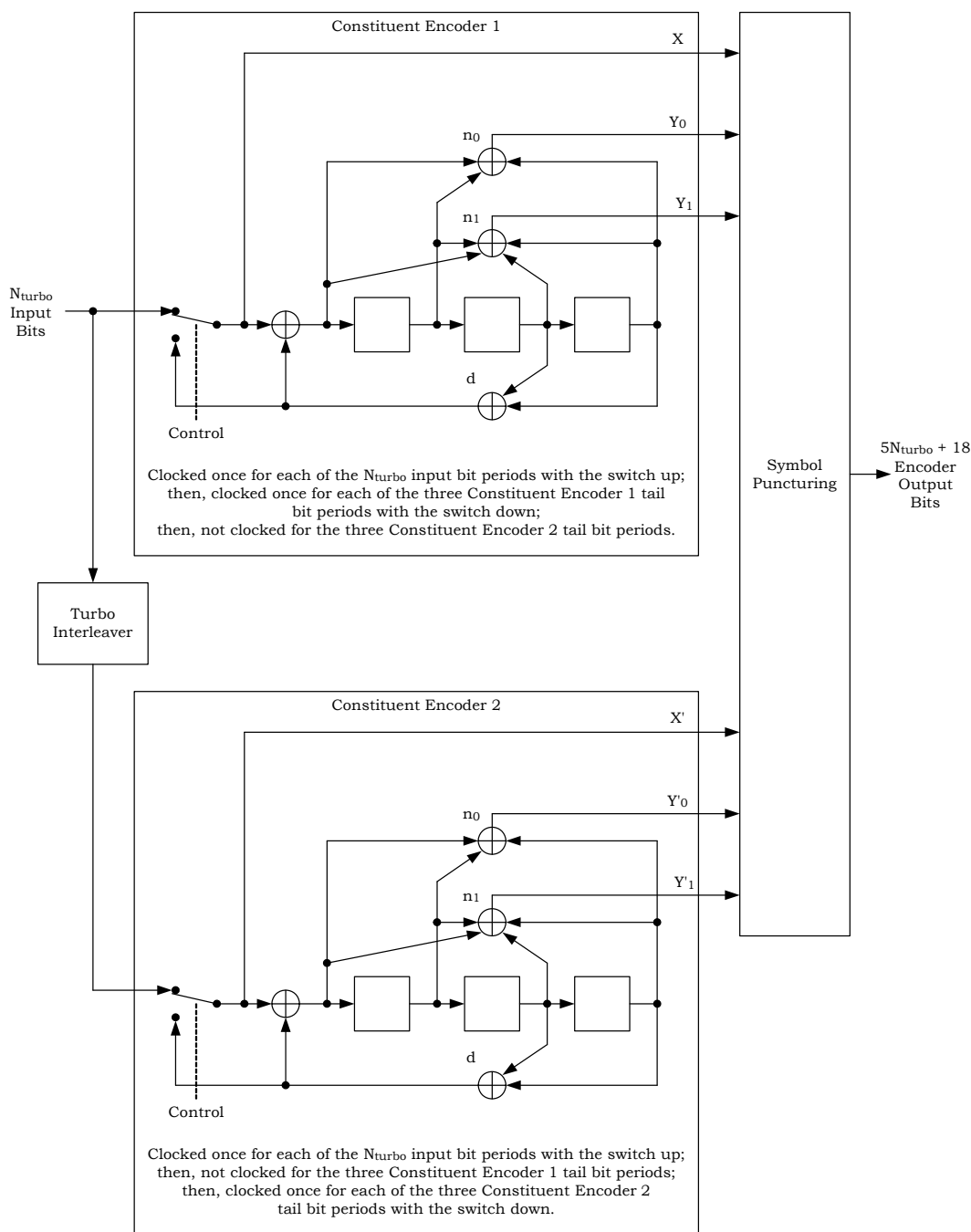


Figure 72—Turbo encoder

9.2.6.3.3.2 Turbo interleavers

The turbo interleaver, which is part of the turbo encoder, shall block interleave the N_{TURBO} input bits.

The turbo interleaver shall be functionally equivalent to an approach where the entire sequence of turbo interleaver input bits are written sequentially into an array at a sequence of addresses, and then the entire sequence is read out from a sequence of addresses that are defined by the procedure described next.

Let the sequence of input addresses be from 0 to $N_{\text{TURBO}} - 1$. Then, the sequence of interleaver output addresses shall be equivalent to those generated by the procedure illustrated in Figure 73 and described as follows:

- Determine the turbo interleaver parameter, n , where n is the smallest integer such that $N_{\text{TURBO}} \leq 2^{n+5}$.
- Initialize an $(n + 5)$ -bit counter to 0.
- Extract the n most significant bits (MSBs) from the counter and add one to form a new value. Then, discard all except the n least significant bits (LSBs) of this value.
- Obtain the n -bit output of the table lookup defined in Table 190 with a read address equal to the five LSBs of the counter. Note that this table depends upon the value of n .
- Multiply the values obtained in steps c) and d), and discard all except the n LSBs.
- Bit-reverse the five LSBs of the counter.
- Form a tentative output address that has its MSBs equal to the value obtained in step f) and its LSBs equal to the value obtained in step e).
- Accept the tentative output address as an output address if it is less than N_{TURBO} ; otherwise, discard it.
- Increment the counter and repeat steps c) through h) until all N_{TURBO} interleaver output addresses are obtained.

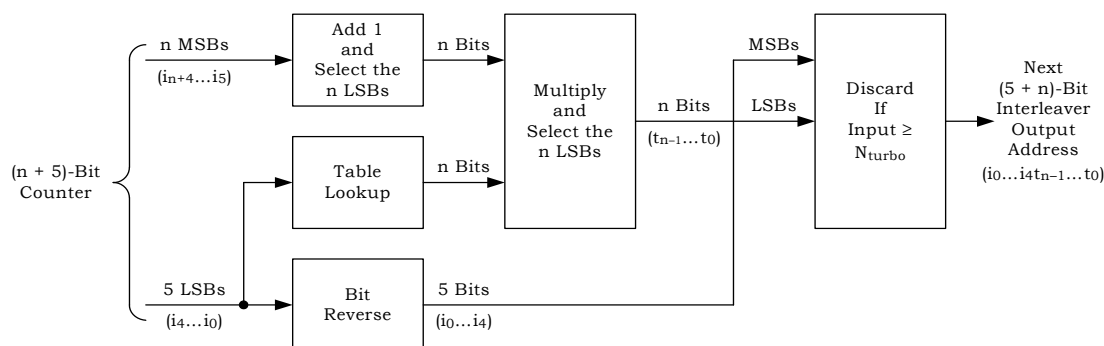


Figure 73—Turbo interleaver output address calculation procedure

Table 190—Turbo interleaver lookup table definition

Table index	$n = 2$ entries	$n = 3$ entries	$n = 4$ entries	$n = 5$ entries	$n = 6$ entries	$n = 7$ entries	$n = 8$ entries	$n = 9$ entries
0	3	1	5	27	3	15	3	13
1	3	1	15	3	27	127	1	335
2	3	3	5	1	15	89	5	87
3	1	5	15	15	13	1	83	15
4	3	1	1	13	29	31	19	15
5	1	5	9	17	5	15	179	1
6	3	1	9	23	1	61	19	333

Table 190—Turbo interleaver lookup table definition (continued)

Table index	n = 2 entries	n = 3 entries	n = 4 entries	n = 5 entries	n = 6 entries	n = 7 entries	n = 8 entries	n = 9 entries
7	1	5	15	13	31	47	99	11
8	1	3	13	9	3	127	23	13
9	1	5	15	3	9	17	1	1
10	3	3	7	15	15	119	3	121
11	1	5	11	3	31	15	13	155
12	1	3	15	13	17	57	13	1
13	1	5	3	1	5	123	3	175
14	1	5	15	13	39	95	17	421
15	3	1	5	29	1	5	1	5
16	3	3	13	21	19	85	63	509
17	1	5	15	19	27	17	131	215
18	3	3	9	1	15	55	17	47
19	3	5	3	3	13	57	131	425
20	3	3	1	29	45	15	211	295
21	1	5	3	17	5	41	173	229
22	3	5	15	25	33	93	231	427
23	1	5	1	29	15	87	171	83
24	3	1	13	9	13	63	23	409
25	1	5	1	13	9	15	147	387
26	3	1	9	23	15	13	243	193
27	1	5	15	13	31	15	213	57
28	3	3	11	13	17	81	189	501
29	1	5	3	1	5	57	51	313
30	1	5	15	13	15	31	15	489
31	3	3	5	13	33	69	67	391

9.2.6.3.4 Low-density parity check encoding

Low-Density Parity Check (LDPC) encoding shall be used to encode the CRC-padded subpackets of the Forward Data Channel and the Forward Superposed Channel if the variable LDPCSupportedFL is set to '1', and if the length of the packet received from the FTC MAC Protocol (see 8.6) is greater than or equal to MaxPacketSizeSixInterlaceFDD or MaxPacketSizeEightInterlaceFDD, MaxPacketSizeTDD44, or MaxPacketSizeTDD63 for the case when the input Forward Data Channel packet corresponds to a six or 8-interlace transmission when DuplexingMode = FDD, a transmission when DuplexingMode = TDD44, and a transmission when DuplexingMode = TDD63, respectively. Here MaxPacketSizeSixInterlaceFDD or

MaxPacketSizeEightInterlaceFDD, MaxPacketSizeTDD44, and MaxPacketSizeTDD63 are configuration attributes of the FTC MAC Protocol (see 8.6). No LDPC encoding shall be used for interlacing structures involving extended transmissions. The FTC MAC Protocol (see 8.6) determines the interlacing structure being used for a given Forward Data Channel packet.

LDPC encoding shall also be used to encode the CRC-padded subpackets of the Reverse OFDMA Data Channel if the variable LDPCSupportedRL is set to '1', and if the length of the packet received from the RTC MAC Protocol (see 8.8) is greater than or equal to MaxRLPacketSize, except if this packet corresponds to an interlacing structure using extended transmissions. No LDPC encoding shall be used for interlacing structures involving extended transmissions. The RTC MAC Protocol (see 8.8) determines whether or not a given packet is transmitted on an interlacing structure using extended transmissions.

9.2.6.3.4.1 Choice of base parity-check matrix

The LDPC code to be used is specified in terms of a base parity-check matrix corresponding to different lifting orders. The lifting order is the size of the binary matrix corresponding to each entry of the base parity matrix. Different base parity-check matrices G_i , $0 \leq i < 6$, are specified in 9.2.6.3.4.5.1. These parity-check matrices represent a lifted LDPC code, i.e., the entries of the matrices are not binary numbers. The matrices consist of positive integers representing shift values, as well as “NULL” locations, which represent a missing edge in the underlying code graph. The interpretation of these matrices as an LDPC code will be described in 9.2.6.3.4.2.

Each matrix G_i has associated values k_B , n_B , s_B , and $L_{\max}$, which are also specified in 9.2.6.3.4.5.1. Here, k_B and n_B determine the size of the matrix G , while $L_{\max}$ denotes the maximum lifting order. The number of columns and rows in G are given by n_B and $n_B - k_B$ respectively. The matrix G_i has associated $k_B = i + 6$. s_B denotes the number of “state columns” in the matrix G_i and is equal to 3 for each of the matrices shown. A state column denotes elements of the codeword that are never transmitted. Each of the specified matrices has a maximum lifting order $L_{\max}$ equal to 1024.

Given the CRC-padded input sub-packet of length k , the lifting value L is chosen as $\log_2 L = \lceil \log_2(k/11) \rceil$. Further, k_B is chosen as $\lceil k/L \rceil$. Note that k_B is at least equal to 6 according to this procedure. Based on this, the matrix index i is chosen as $i = k_B - 6 = \lceil k/L \rceil - 6$.

Henceforth, the index i will be dropped and the matrix G_i is referred to only as G .

9.2.6.3.4.2 Generation of the parity-check matrix

The base matrix G chosen in the previous subclause shall be converted to a new base matrix G' , corresponding to the actual lifting order L rather than the maximal lifting order $L_{\max} = 1024$. The matrix G' has the same size as G . An entry g' in G' shall be determined from the entry g at the same location in G according to the formula $g' = \lfloor gL / L_{\max} \rfloor$. NULL locations in G shall remain NULL locations in G' .

The matrix G' shall further be converted to a matrix G'' with twice the number of rows and columns as in G' . This shall be done by replacing each non-NULL entry g' in G' by a 2×2 matrix according to the following procedure:

- If g' is even, replace g' by a 2×2 matrix with first row being given by $[g'/2, \text{“NULL”}]$ and the second row being given by $[\text{“NULL”}, g'/2]$.
- If g' is odd, replace g' by a 2×2 matrix with the first row being given by $[\text{“NULL”}, (g' + 1)/2]$ and the second row being given by $[(g' - 1)/2, \text{“NULL”}]$.

NULL locations in G' shall be replaced by a 2×2 matrix containing entirely of NULL locations in G'' . The matrix G'' is the base parity-check matrix of size $(2(n_B - k_B), 2 n_B)$ with a lifting order of $L/2$.

The base matrix G'' shall be converted to a base matrix G''' by applying permutation P_i to the columns of G'' and permutation Q_i to the rows of G'' , where the permutations P_i and Q_i are described in 9.2.6.3.4.1. The subscript i in P_i and Q_i refers to the subscript $G_i = G$ and thus takes values in $0, \dots, 5$. The numbers in the third column (Q_i) and fourth column (P_i) of Table 191 indicate row index and column index of G'' corresponding to i . Specifically, the permutations P_i and Q_i are applied such that the element in G''' with row index k_r and column index k_c is the same as the element in G'' with row index $P_i(k_r)$ and column index $Q_i(k_c)$.

The first (leftmost) $2k_B$ columns of G''' correspond to the information bits V^{in} and $(k_B L - k)$ zero-padded bits. The subsequent K_i columns (K_i depends on G_i) together with the first K_i rows form a lifted parity-check matrix that consist of a degree 3 variable node (i.e., a column with three non-NULL elements) followed by $K_i - 1$ degree 2 variable nodes. The degree 2 parity nodes form a dual-diagonal structure and the degree 3 variable node closes the loop of the dual-diagonal structure. Each non-NULL entry of degree 2 variable node in the dual diagonal structure has the lifting parameter zero, corresponding to identity matrix, on both edges. The loop closing edges on the degree 3 node have the same lifting value “a”. The non-loop edge of the degree 3 node has lifting parameter zero corresponding to identity matrix so the lifting structure of this degree 3 node is “a-0-a”. The remaining columns in G''' are degree 1 variable nodes. The base matrices G_i , $i = 0, \dots, 5$ each contain, up to permutation, a dual diagonal structure, a loop closing degree 3 encoding node and a loop closing degree 2 encoding node (excluding any edges connected to constraints that are in turn connected to degree 1 variable nodes). As the scaled base graph G' is transformed into the expanded graph G'' and permuted into G''' , each of the degree 2 and degree 3 loop closing nodes of G' generates an information node and an encoding node of the same degree, such that the encoding node is part of the core encoding structure of G''' described above. The loop closing degree 2 variable node of G_i has cyclic shift 0 and $L_{max} - 1$. The corresponding loop-closing edges of the degree 3 variable node have cyclic shifts, which are $3L_{max}/4 - 1$ and $3L_{max}/4$. The third non-loop edge of the degree 3 node has cyclic shift 0. As the graph G_i is scaled to generate the graph G' , with values of L from 16 to 1024, the scaled base matrices G' inherit the same structure on the loop-closing degree 2 and degree 3 nodes, with L_{max} replaced by L . This induces the lifting value of 0 on the loop closing degree 2 information node in G''' , and the “a-0-a” lifting on the loop closing degree 3 encoding node of G''' , where $a = 3L/8$.

An example for the generation of G' , G'' , G''' is shown in 9.2.6.3.4.5.2.

The base matrix G''' shall be converted to a binary parity-check matrix H''' by replacing each non-NULL entry in G''' by a $L/2 \times L/2$ square matrix with binary entries. An entry g''' in G''' shall be replaced by a cyclic shift matrix with parameter g''' . The cyclic shift matrix with parameter g''' is defined as the matrix whose value in the location (i, j) is given by '1' if $(i - j) \bmod L/2 = g'''$, and is given by '0' otherwise. Here, the location (i, j) denotes the i^{th} row and j^{th} column. NULL locations in G''' shall be replaced by an $L/2 \times L/2$ all-zeros matrix.

The CRC-padded input sub-packet of length k shall be extended to length $k_B L$ by inserting in the packet $z_p = k_B L - k$ zeros so that the resulting packet has length $k_B L$. Denote again the original CRC-padded input sub-packet by V^{in} and denote the zero-padded input by a column vector $V^I = (V_0^I, V_1^I, \dots, V_{k'-1}^I)$ where $k' = k_B L$. The locations of the zeros in V^I are as follows. If V^I is partitioned into $2k_B$ blocks of size $L/2$, then the zeros are inserted at the ends of blocks $2k_B - 4$ and $2k_B - 3$. Each block has an equal number of zeros if z_p is even and block $2k_B - 3$ has one more than block $2k_B - 4$ if z_p is odd.

More precisely, define $z_p' = \lfloor z_p/2 \rfloor$ and $z_p'' = \lceil z_p/2 \rceil$. Let the notation V_i^I and V_i^{in} denote the i^{th} element of V^I and V^{in} respectively. The elements of the vector V^I are given by the following:

$$V_i^I = V_i^{in} \text{ for } i < (2k_B - 3)(L/2) - z_p'.$$

$$V_i^I = 0 \text{ for } (2k_B - 3)(L/2) - z_p' \leq i < (2k_B - 3)(L/2)$$

$$V_i^L = V_{i-zp}^{\text{in}}, \text{ for } (2k_B - 3)(L/2) \leq i < (2k_B - 2)(L/2) - zp''$$

$$V_i^L = 0 \text{ for } (2k_B - 2)(L/2) - zp'' \leq i < (2k_B - 2)(L/2)$$

$$V_i^L = V_{i-zp}^{\text{in}} \text{ for } i \geq (2k_B - 2)(L/2)$$

A vector V''' of length $n_B L$ shall be defined as the vector that satisfies the following conditions:

$H'''V''' = 0$, where the matrix multiplication $H'''V'''$ is over the binary field.

The first $k_B L$ entries of V''' are the same as the entries of V^L .

The vector V''' is of length $n_B L$ and may therefore be viewed as the concatenation of $2n_B$ subsequences each of length $L/2$.

A binary sequence V'' shall be obtained from V''' by permuting the order of sequences of V''' according to the inverse of the permutation P_i .

A binary sequence V^O of length $n_B L$ shall be obtained from V'' by bit-wise interleaving pairs of subsequences from V'' . More specifically,

$$V_{jL+j'}^O = V''_{jL+(L/2)(j' \bmod 2) + \lfloor j'/2 \rfloor} \text{ where } j = 0, 1, \dots, n_B - 1 \text{ and } j' = 0, 1, \dots, L - 1$$

The LDPC output vector V^{out} of length $n = L n_B - s_B L - (k_B L - k)$ shall be obtained as a subsequence of V^O by deleting the zero padding and state variables from V^O . The permutations P_i are such that the zero-padded bits appear contiguously in V^O in positions $k - L$ to $(k_B - 1)L - 1$ and the state variables appear as the first $s_B L$ bits. Note that $s_B = 3$ in all cases. Thus

$$V_i^{\text{out}} = V_{i+3L}^O \text{ for } 0 \leq i < k - 4L$$

$$V_i^{\text{out}} = V_{i+3L+zp}^O \text{ for } k - 4L \leq i < (n_B - 3)L - zp$$

Table 191—Permutation patterns for the construction of dual diagonal structure

i	K _i	Q _i	P _i
0	12	11 5 3 1 7 9 10 4 2 0 6 8	0 1 2 3 4 5 6 7 8 9 10 12 13 15 21 19 17 41 11 14 20 18 16 40 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
1	16	7 1 13 15 11 5 9 3 6 0 12 14 10 4 8 2	0 1 2 3 4 5 6 7 8 9 10 11 12 14 15 17 23 51 21 49 19 43 13 16 22 50 20 48 18 42 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 44 45 46 47 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
2	14	11 3 1 13 9 5 7 10 2 0 12 8 4 6	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 16 17 19 25 23 59 21 41 15 18 24 22 58 20 40 26 27 28 29 30 31 32 33 34 35 36 37 38 39 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85
3	18	17 15 13 9 11 5 7 1 3 16 14 12 8 10 4 6 0 2	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 18 19 21 53 27 51 25 47 23 43 17 20 52 26 50 24 46 22 42 28 29 30 31 32 33 34 35 36 37 38 39 40 41 44 45 48 49 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95

Table 191—Permutation patterns for the construction of dual diagonal structure (continued)

i	K _i	Q _i	P _i
4	18	5 17 15 11 13 9 3 7 1 4 16 14 10 12 8 2 6 0	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 21 23 53 29 49 27 47 25 45 19 22 52 28 48 26 46 24 44 30 31 32 33 34 35 36 37 38 39 40 41 42 43 50 51 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105
5	18	11 9 17 15 7 13 5 1 3 10 8 16 14 6 12 4 0 2	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 22 23 25 55 31 53 29 51 27 41 21 24 54 30 52 28 50 26 40 32 33 34 35 36 37 38 39 42 43 44 45 46 47 48 49 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115

9.2.6.3.4.3 Encoding

In this subclause an efficient encoding method is presented according to which the sequence V'' , as defined in 9.2.6.3.4.2, is computed from V^1 . The method will describe a procedure to generate the sequence V'' . Recall that in 9.2.6.3.4.2 it was described how to produce the matrix G'' from the matrix G (which is also G_j). Efficient encoding of V^1 to a sequence V'' satisfying $H''V'' = 0$ is described.

The computation of V'' given V^1 is particularly simple due to the structure H'' inherits from G'' . The parity-check matrix H'' lifted from G'' takes the form of Equation (20).

$$M_1 = \begin{bmatrix} M_1 & 0 \\ M_2 & I \end{bmatrix} \quad (20)$$

where

$$M_1 = \begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix} \text{ is a } (L/2)K_i \times (L/2)(2k_B + K_i) \text{ matrix}$$

T is lower triangular

$$\begin{bmatrix} B & T \\ D & E \end{bmatrix} \text{ is invertible}$$

D is $L/2 \times L/2$.

The encoding procedure is composed of two stages. Let $c = (s, p_1, p_2, p_3)$ be a codeword where s denotes systematic part, p_1, p_2 , and p_3 are parity parts. In the first stage, a part of codeword p_1, p_2 is obtained using M_1 depending on the systematic information s . In the second stage, the remaining part of the codeword p_3 is obtained by simple single parity-check coding using $[M_2 I]$. The whole procedure for encoding is as follows.

- Obtain $\begin{bmatrix} B & T \\ \phi & 0 \end{bmatrix}$ from Gaussian elimination on $\begin{bmatrix} B & T \\ D & E \end{bmatrix}$, where $\phi = ET^{-1}B + D = 1$.
- Compute As^T and Cs^T
- Compute $y = T^{-1}As^T$
- Compute $p_1^T = Ey + Cs^T$
- Compute p_2^T using $p_2^T = T^{-1}(As^T + Bp_1^T)$
- Compute p_3^T by single parity-check coding using $[M_2 I]$

A sequence V^{in} satisfying $H^{\text{in}}V^{\text{in}}=0$ is obtained from 1–6. The sequence V^{out} is then obtained from V^{in} by permutation as described in 9.2.6.3.4.2.

9.2.6.3.4.4 Truncation

For the Forward Data Channel and Forward Superposed Data Channel packets, the truncation operation shall be carried out as described as follows:

- The truncation operation depends on the packet size $N_{\text{PACKET\_BITS}}$ of the packet received from the FTC MAC Protocol (see 8.6), and the variables $\text{MaxRateOneFifthPacketSize}$, $\text{MaxRateOneThirdPacketSize}$ and $\text{MaxRateOneHalfPacketSize}$. $\text{MaxRateOneFifthPacketSize}$ is equal to one of the parameters $\text{MaxRateOneFifthPacketSizeEightInterlaceFDDLDPC}$, $\text{MaxRateOneFifthPacketSizeSixInterlaceFDDLDPC}$, $\text{MaxRateOneFifthPacketSizeTDD44LDPC}$, or $\text{MaxRateOneFifthPacketSizeTDD63LDPC}$ depending on whether the Forward Data Channel or the Forward Superposed Data Channel packet is transmitted using an 8-interlace HARQ structure or a 6-interlace HARQ structure when $\text{DuplexingMode} = \text{FDD}$, or is transmitted when $\text{DuplexingMode} = \text{TDD44}$ or is transmitted when $\text{DuplexingMode} = \text{TDD63}$. $\text{MaxRateOneThirdPacketSize}$ is equal to one of the parameters $\text{MaxRateOneThirdPacketSizeEightInterlaceFDDLDPC}$, $\text{MaxRateOneThirdPacketSizeSixInterlaceFDDLDPC}$, $\text{MaxRateOneThirdPacketSizeTDD44LDPC}$, and $\text{MaxRateOneThirdPacketSizeTDD63LDPC}$ depending on whether the Forward Data Channel or the Forward Superposed Data Channel packet is transmitted using an 8-interlace HARQ structure or a 6-interlace HARQ structure when $\text{DuplexingMode} = \text{FDD}$, or is transmitted when $\text{DuplexingMode} = \text{TDD44}$ or is transmitted when $\text{DuplexingMode} = \text{TDD63}$. $\text{MaxRateOneHalfPacketSize}$ is equal to one of the parameters $\text{MaxRateOneHalfPacketSizeEightInterlaceFDDLDPC}$, $\text{MaxRateOneHalfPacketSizeSixInterlaceFDDLDPC}$, $\text{MaxRateOneHalfPacketSizeTDD44LDPC}$, and $\text{MaxRateOneHalfPacketSizeTDD63LDPC}$ depending on whether the Forward Data Channel or the Forward Superposed Data Channel packet is transmitted using an 8-interlace HARQ structure or a 6-interlace HARQ structure when $\text{DuplexingMode} = \text{FDD}$, or is transmitted when $\text{DuplexingMode} = \text{TDD44}$ or is transmitted when $\text{DuplexingMode} = \text{TDD63}$. The FTC MAC Protocol (see 8.6) determines which HARQ interlacing structure is used for transmitting the Forward Data Channel or the Forward Superposed Data Channel packet.
- $\text{MaxRateOneFifthPacketSizeEightInterlaceFDDLDPC}$, $\text{MaxRateOneFifthPacketSizeSixInterlaceFDDLDPC}$, $\text{MaxRateOneFifthPacketSizeTDD44LDPC}$, $\text{MaxRateOneFifthPacketSizeTDD63LDPC}$, $\text{MaxRateOneThirdPacketSizeEightInterlaceFDDLDPC}$, $\text{MaxRateOneThirdPacketSizeSixInterlaceFDDLDPC}$, $\text{MaxRateOneThirdPacketSizeTDD44LDPC}$, $\text{MaxRateOneThirdPacketSizeTDD63LDPC}$, $\text{MaxRateOneHalfPacketSizeEightInterlaceFDDLDPC}$, $\text{MaxRateOneHalfPacketSizeSixInterlaceFDDLDPC}$, $\text{MaxRateOneHalfPacketSizeTDD44LDPC}$, and $\text{MaxRateOneHalfPacketSizeTDD63LDPC}$ are configuration attributes of the FTC MAC Protocol (see 8.6).
- If $N_{\text{PACKET\_BITS}} < \text{MaxRateOneFifthPacketSize}$, the sequence V^{out} is not truncated.
- If $\text{MaxRateOneFifthPacketSize} \leq N_{\text{PACKET\_BITS}} < \text{MaxRateOneThirdPacketSize}$, the sequence V^{out} is truncated to length $3k$, i.e., all elements with indices greater than or equal to $3k$ are deleted.
- If $\text{MaxRateOneThirdPacketSize} \leq N_{\text{PACKET\_BITS}} < \text{MaxRateOneHalfPacketSize}$, the sequence V^{out} is truncated to length $2L$, i.e., all elements with indices greater than or equal to $2L$ are deleted.
- If $\text{MaxRateOneHalfPacketSize} \leq N_{\text{PACKET\_BITS}}$, the sequence V^{out} is truncated to length $3k/2$, i.e., all elements with indices greater than or equal to $2k$ are deleted.
- The output of the truncation operation is denoted by V^{tr} .

For Reverse OFDMA Data Channel packets, the output V^{tr} of the truncation operation shall be equal to V^{out} .

The sequence V^{tr} is a truncated version of the permuted sequence V^{out} and is in the order of bit transmission, i.e., the different HARQ transmissions of this packet are generated in the order specified by V^{tr} .

9.2.6.3.4.5 Parity-check matrices for the LDPC code

The various base parity-check matrices G_i are tabulated in this subclause.

9.2.6.3.4.5.1 Base parity-check matrices

The parity-check matrix of LDPC codes is represented by degrees, positions and shift values of non-NULL elements in the base matrix. The second column in the table provides the degree of each row in the base parity-check matrix, which is defined as the number of non-NULL elements in that row. The third column in the table provides the column indices of the non-NULL elements in that row, while the last column provides the integer entries corresponding to the non-NULL locations. These integer entries denote a cyclic shift value modulo L_{max} .

Table 192 specifies the matrix G_0 with the following parameters: $k_B = 6$, $n_B = 33$, $s_B = 3$, and $L_{\text{max}} = 1024$.

Table 192—Base matrix G_0

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
0	6	1, 2, 3, 4, 8, 9	110, 680, 424, 180, 0, 0
1	6	0, 2, 3, 6, 9, 10	702, 768, 863, 0, 0, 0
2	6	0, 1, 3, 4, 7, 10	360, 259, 652, 753, 0, 0
3	4	1, 4, 8, 20	402, 948, 0, 0
4	4	0, 5, 6, 20	318, 0, 767, 0
5	4	2, 5, 6, 7	154, 1023, 768, 0
6	3	0, 1, 11	885, 323, 0
7	3	0, 2, 12	617, 220, 0
8	4	1, 2, 3, 13	799, 519, 669, 0
9	4	0, 1, 4, 14	900, 72, 669, 0
10	4	0, 2, 6, 15	574, 253, 352, 0
11	4	1, 2, 7, 16	848, 280, 920, 0
12	4	0, 1, 5, 17	548, 928, 355, 0
13	4	0, 2, 3, 18	17, 376, 147, 0
14	4	0, 1, 4, 19	795, 823, 473, 0
15	4	0, 2, 8, 21	519, 424, 712, 0
16	3	1, 6, 22	952, 449, 0
17	3	2, 7, 23	887, 798, 0
18	4	0, 1, 9, 24	256, 93, 348, 0
19	3	2, 3, 25	492, 856, 0

Table 192—Base matrix G_0 (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
20	4	1, 2, 10, 26	589, 1016, 705, 0
21	3	0, 4, 27	26, 166, 0
22	4	1, 2, 5, 28	525, 584, 845, 0
23	3	0, 8, 29	10, 331, 0
24	3	1, 9, 30	125, 310, 0
25	3	2, 10, 31	239, 641, 0
26	4	0, 1, 6, 32	557, 609, 448, 0

Table 193 specifies the matrix G_1 with the following parameters: $k_B = 7$, $n_B = 38$, $s_B = 3$, and $L_{\max} = 1024$.

Table 193—Base matrix G_1

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
0	6	0, 1, 3, 4, 8, 11	556, 1023, 480, 944, 0, 0
1	5	0, 5, 6, 7, 21	430, 916, 0, 767, 0
2	5	1, 3, 4, 9, 24	295, 907, 87, 0, 0
3	5	2, 4, 6, 7, 8	809, 501, 1023, 768, 0
4	4	1, 3, 9, 21	954, 710, 0, 0
5	4	2, 5, 10, 24	558, 360, 0, 0
6	4	2, 7, 11, 25	275, 0, 0, 0
7	4	0, 5, 10, 25	935, 568, 0, 0
8	3	0, 1, 12	195, 989, 0
9	3	0, 2, 13	550, 728, 0
10	4	1, 2, 3, 14	532, 26, 698, 0
11	4	0, 1, 4, 15	664, 862, 709, 0
12	4	0, 2, 5, 16	938, 440, 978, 0
13	4	1, 2, 8, 17	394, 995, 17, 0
14	4	0, 3, 7, 18	538, 175, 117, 0
15	4	1, 2, 6, 19	428, 105, 929, 0
16	4	0, 1, 9, 20	30, 264, 832, 0
17	4	0, 2, 10, 22	514, 410, 978, 0
18	4	1, 2, 4, 23	487, 249, 204, 0
19	4	0, 1, 11, 26	526, 126, 906, 0

Table 193—Base matrix G_1 (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
20	4	0, 2, 5, 27	10, 90, 889, 0
21	3	1, 7, 28	126, 714, 0
22	3	2, 8, 29	312, 967, 0
23	4	0, 3, 6, 30	954, 302, 63, 0
24	3	0, 9, 31	33, 624, 0
25	4	0, 3, 4, 32	524, 752, 227, 0
26	3	1, 5, 33	647, 346, 0
27	3	2, 10, 34	918, 602, 0
28	4	0, 3, 4, 35	14, 131, 816, 0
29	3	1, 7, 36	216, 103, 0
30	3	2, 11, 37	893, 771, 0

Table 194 specifies the matrix G_2 with the following parameters: $k_B = 8$, $n_B = 43$, $s_B = 3$, and $L_{\max} = 1024$.

Table 194—Base matrix G_2

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
0	7	0, 2, 3, 5, 6, 11, 12	719, 328, 939, 579, 765, 0, 0
1	7	0, 1, 3, 4, 5, 9, 12	955, 1019, 365, 503, 882, 0, 0
2	5	1, 4, 5, 10, 20	495, 720, 413, 0, 0
3	5	0, 6, 7, 8, 20	63, 163, 0, 767, 0
4	5	1, 3, 4, 10, 29	629, 319, 818, 0, 0
5	5	2, 6, 7, 8, 9	247, 412, 1023, 768, 0
6	4	2, 8, 11, 29	928, 0, 0, 0
7	3	0, 1, 13	32, 190, 0
8	3	0, 2, 14	243, 596, 0
9	4	1, 2, 3, 15	880, 833, 329, 0
10	4	0, 1, 4, 16	224, 840, 208, 0
11	4	2, 5, 9, 17	479, 222, 17, 0
12	4	0, 1, 6, 18	296, 856, 651, 0
13	4	2, 3, 8, 19	926, 211, 167, 0
14	4	0, 7, 13, 21	764, 166, 387, 0
15	4	1, 2, 10, 22	238, 925, 405, 0

Table 194—Base matrix G_2 (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
16	4	3, 4, 14, 23	850, 922, 852, 0
17	4	0, 1, 5, 24	412, 96, 627, 0
18	4	0, 2, 11, 25	536, 443, 773, 0
19	4	1, 2, 12, 26	551, 91, 400, 0
20	4	0, 3, 6, 27	567, 242, 205, 0
21	4	1, 2, 8, 28	556, 157, 27, 0
22	3	0, 9, 30	10, 886, 0
23	4	1, 2, 20, 31	831, 252, 11, 0
24	4	0, 3, 4, 32	755, 623, 867, 0
25	3	1, 7, 33	608, 72, 0
26	4	0, 2, 10, 34	60, 516, 772, 0
27	4	1, 3, 5, 35	289, 906, 292, 0
28	3	2, 6, 36	600, 48, 0
29	3	0, 11, 37	565, 458, 0
30	4	0, 4, 8, 38	428, 6, 413, 0
31	3	1, 12, 39	958, 131, 0
32	3	2, 5, 40	577, 146, 0
33	4	1, 4, 6, 41	734, 257, 619, 0
34	3	2, 8, 42	612, 634, 0

Table 195 specifies the matrix G_3 with the following parameters: $k_B = 9$, $n_B = 48$, $s_B = 3$, and $L_{\max} = 1024$.

Table 195—Base matrix G_3

Row index	Row degree	Column positions of non-NULL elements in row	Shift Numbers of non-NULL elements in row
0	5	1, 3, 5, 11, 21	854, 545, 457, 0, 0
1	5	0, 6, 8, 9, 21	282, 1001, 0, 767, 0
2	5	2, 5, 6, 12, 23	984, 677, 794, 0, 0
3	5	1, 4, 7, 11, 23	169, 618, 313, 0, 0
4	5	2, 3, 6, 13, 25	129, 699, 370, 0, 0
5	5	0, 4, 9, 12, 25	156, 934, 0, 0, 0
6	5	1, 4, 7, 13, 26	923, 840, 117, 0, 0
7	5	0, 3, 5, 10, 26	538, 243, 83, 0, 0

Table 195—Base matrix G_3 (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift Numbers of non-NULL elements in row
8	5	2, 7, 8, 9, 10	464, 1009, 1023, 768, 0
9	3	0, 1, 14	259, 434, 0
10	4	0, 1, 2, 15	274, 901, 1004, 0
11	4	0, 2, 14, 16	640, 997, 988, 0
12	4	1, 2, 10, 17	683, 54, 385, 0
13	4	0, 3, 4, 18	679, 253, 646, 0
14	4	1, 5, 6, 19	47, 418, 332, 0
15	4	2, 7, 9, 20	343, 26, 175, 0
16	4	0, 8, 15, 22	514, 671, 496, 0
17	4	1, 2, 11, 24	972, 433, 993, 0
18	4	0, 1, 12, 27	235, 223, 885, 0
19	4	0, 2, 13, 28	555, 943, 892, 0
20	4	1, 2, 21, 29	696, 574, 233, 0
21	4	0, 1, 23, 30	975, 510, 815, 0
22	4	0, 3, 4, 31	5, 818, 898, 0
23	3	2, 5, 32	350, 159, 0
24	4	0, 3, 6, 33	36, 397, 807, 0
25	4	1, 3, 7, 34	492, 502, 467, 0
26	3	1, 9, 35	162, 631, 0
27	4	2, 3, 10, 36	608, 944, 599, 0
28	3	2, 8, 37	394, 630, 0
29	3	0, 11, 38	48, 576, 0
30	4	1, 4, 10, 39	952, 521, 455, 0
31	3	2, 12, 40	304, 300, 0
32	4	0, 4, 5, 41	982, 602, 915, 0
33	3	1, 6, 42	740, 710, 0
34	4	0, 4, 7, 43	783, 491, 307, 0
35	3	1, 13, 44	431, 275, 0
36	4	2, 3, 5, 45	802, 46, 83, 0
37	3	2, 9, 46	556, 239, 0
38	3	4, 6, 47	812, 72, 0

Table 196 specifies the matrix G_4 with the following parameters: $k_B = 10$, $n_B = 53$, $s_B = 3$, and $L_{\max} = 1024$.

Table 196—Base matrix G_4

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
0	6	0, 5, 6, 9, 10, 22	892, 146, 990, 0, 767, 0
1	6	1, 3, 4, 5, 12, 23	350, 577, 321, 15, 0, 0
2	6	2, 4, 7, 9, 10, 11	242, 771, 989, 1023, 768, 0
3	5	1, 3, 8, 12, 22	627, 183, 532, 0, 0
4	5	2, 7, 10, 13, 23	614, 847, 0, 0, 0
5	5	2, 7, 8, 14, 24	456, 923, 264, 0, 0
6	5	0, 5, 6, 13, 24	664, 365, 587, 0, 0
7	5	1, 3, 6, 14, 26	767, 882, 392, 0, 0
8	5	0, 4, 8, 11, 26	124, 908, 915, 0, 0
9	3	0, 1, 15	815, 184, 0
10	4	0, 1, 2, 16	476, 804, 646, 0
11	4	0, 2, 3, 17	6, 735, 10, 0
12	4	1, 2, 11, 18	512, 95, 710, 0
13	4	0, 4, 5, 19	949, 94, 860, 0
14	4	1, 6, 7, 20	191, 452, 860, 0
15	4	2, 8, 10, 21	455, 231, 802, 0
16	4	9, 15, 16, 25	489, 984, 736, 0
17	4	0, 1, 12, 27	555, 536, 777, 0
18	4	2, 3, 13, 28	527, 612, 534, 0
19	4	0, 1, 14, 29	21, 227, 461, 0
20	4	2, 3, 22, 30	897, 119, 618, 0
21	4	0, 4, 17, 31	530, 581, 453, 0
22	4	1, 2, 5, 32	977, 76, 139, 0
23	4	0, 4, 6, 33	407, 302, 832, 0
24	4	1, 2, 23, 34	616, 233, 419, 0
25	4	0, 3, 7, 35	294, 500, 831, 0
26	3	1, 8, 36	994, 254, 0
27	4	2, 3, 24, 37	278, 1001, 589, 0
28	4	0, 3, 10, 38	4, 69, 141, 0
29	4	1, 3, 26, 39	956, 629, 420, 0

Table 196—Base matrix G_4 (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
30	3	2, 11, 40	422, 541, 0
31	4	0, 3, 9, 41	816, 663, 475, 0
32	4	1, 3, 12, 42	349, 1010, 663, 0
33	3	2, 13, 43	94, 922, 0
34	4	0, 4, 5, 44	354, 776, 356, 0
35	3	1, 6, 45	995, 494, 0
36	4	2, 4, 7, 46	271, 911, 178, 0
37	3	0, 8, 47	7, 393, 0
38	4	1, 4, 10, 48	535, 888, 24, 0
39	3	2, 14, 49	854, 792, 0
40	3	4, 5, 50	792, 143, 0
41	3	3, 22, 51	170, 761, 0
42	4	3, 4, 6, 52	427, 900, 106, 0

Table 197 specifies the matrix G_5 with the following parameters: $k_B = 11$, $n_B = 58$, $s_B = 3$, and $L_{\max} = 1024$.

Table 197—Base matrix G_5

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
0	6	1, 4, 8, 9, 13, 20	572, 998, 930, 221, 0, 0
1	6	0, 5, 6, 10, 11, 20	340, 579, 789, 0, 767, 0
2	6	1, 3, 6, 7, 13, 25	368, 46, 358, 978, 0, 0
3	6	0, 4, 7, 9, 14, 26	921, 128, 589, 32, 0, 0
4	6	0, 4, 5, 8, 12, 27	757, 230, 714, 823, 0, 0
5	6	2, 3, 7, 10, 11, 12	729, 583, 1, 1023, 768, 0
6	5	2, 5, 8, 14, 25	997, 211, 438, 0, 0
7	5	2, 3, 11, 15, 26	418, 127, 0, 0, 0
8	5	1, 6, 9, 15, 27	598, 570, 943, 0, 0
9	3	0, 1, 16	847, 386, 0
10	4	0, 1, 2, 17	371, 107, 268, 0
11	4	0, 2, 3, 18	119, 226, 30, 0
12	4	1, 2, 12, 19	355, 74, 27, 0
13	4	4, 5, 6, 21	310, 74, 15, 0

Table 197—Base matrix G_5 (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
14	4	0, 7, 8, 22	394, 952, 73, 0
15	4	1, 2, 9, 23	55, 95, 768, 0
16	4	10, 11, 16, 24	912, 973, 727, 0
17	4	0, 13, 17, 28	697, 518, 709, 0
18	4	1, 3, 18, 29	226, 568, 330, 0
19	4	2, 4, 14, 30	41, 113, 948, 0
20	4	0, 3, 15, 31	31, 698, 549, 0
21	4	1, 4, 5, 32	262, 256, 243, 0
22	4	0, 2, 20, 33	574, 60, 651, 0
23	4	1, 5, 6, 34	946, 464, 986, 0
24	4	2, 6, 7, 35	684, 243, 20, 0
25	4	0, 3, 8, 36	541, 250, 136, 0
26	4	1, 2, 9, 37	265, 559, 896, 0
27	4	0, 4, 11, 38	699, 43, 320, 0
28	4	0, 1, 25, 39	24, 47, 193, 0
29	3	2, 26, 40	479, 995, 0
30	4	0, 5, 7, 41	44, 855, 57, 0
31	4	1, 2, 27, 42	150, 720, 179, 0
32	4	1, 3, 8, 43	168, 140, 985, 0
33	3	2, 12, 44	689, 429, 0
34	4	0, 4, 6, 45	515, 385, 75, 0
35	4	1, 5, 10, 46	75, 396, 1017, 0
36	3	0, 13, 47	31, 388, 0
37	4	1, 6, 7, 48	226, 123, 612, 0
38	3	2, 14, 49	509, 38, 0
39	4	2, 7, 8, 50	726, 430, 767, 0
40	3	3, 8, 51	838, 277, 0
41	3	4, 9, 52	383, 745, 0
42	3	3, 5, 53	772, 267, 0
43	3	6, 9, 54	270, 297, 0
44	3	4, 7, 55	743, 598, 0

Table 197—Base matrix G_5 (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
45	3	5, 12, 56	66, 373, 0
46	4	3, 6, 8, 57	455, 973, 737, 0

9.2.6.3.4.5.2 Generation of the Matrices G_i' , G_i'' , G_i'''

The following example shows the procedure for converting the matrix G_0 to the matrices G_0' , G_0'' , and G_0''' .

Table 198 represents the matrix G_0' with a lifting size of 512, $k_B = 6$, $n_B = 33$, $s_B = 3$, and $L_{\max} = 1024$.

Table 198—Base matrix G_0'

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
0	6	1, 2, 3, 4, 8, 9	55, 340, 212, 90, 0, 0
1	6	0, 2, 3, 6, 9, 10	351, 384, 431, 0, 0, 0
2	6	0, 1, 3, 4, 7, 10	180, 129, 326, 376, 0, 0
3	4	1, 4, 8, 20	201, 474, 0, 0
4	4	0, 5, 6, 20	159, 0, 383, 0
5	4	2, 5, 6, 7	77, 511, 384, 0
6	3	0, 1, 11	442, 161, 0
7	3	0, 2, 12	308, 110, 0
8	4	1, 2, 3, 13	399, 259, 334, 0
9	4	0, 1, 4, 14	450, 36, 334, 0
10	4	0, 2, 6, 15	287, 126, 176, 0
11	4	1, 2, 7, 16	424, 140, 460, 0
12	4	0, 1, 5, 17	274, 464, 177, 0
13	4	0, 2, 3, 18	8, 188, 73, 0
14	4	0, 1, 4, 19	397, 411, 236, 0
15	4	0, 2, 8, 21	259, 212, 356, 0
16	3	1, 6, 22	476, 224, 0
17	3	2, 7, 23	443, 399, 0
18	4	0, 1, 9, 24	128, 46, 174, 0
19	3	2, 3, 25	246, 428, 0
20	4	1, 2, 10, 26	294, 508, 352, 0
21	3	0, 4, 27	13, 83, 0

Table 198—Base matrix G_0' (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
22	4	1, 2, 5, 28	262, 292, 422, 0
23	3	0, 8, 29	5, 165, 0
24	3	1, 9, 30	62, 165, 0
25	3	2, 10, 31	119, 320, 0
26	4	0, 1, 6, 32	278, 304, 224, 0

Table 199 represents the matrix G_0'' with a lifting size of 256.

Table 199—Base matrix G_0''

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
0	6	3, 4, 6, 8, 16, 18	28, 170, 106, 45, 0, 0
1	6	2, 5, 7, 9, 17, 19	27, 170, 106, 45, 0, 0
2	6	1, 4, 7, 12, 18, 20	176, 192, 216, 0, 0, 0
3	6	0, 5, 6, 13, 19, 21	175, 192, 215, 0, 0, 0
4	6	0, 3, 6, 8, 14, 20	90, 65, 163, 188, 0, 0
5	6	1, 2, 7, 9, 15, 21	90, 64, 163, 188, 0, 0
6	4	3, 8, 16, 40	101, 237, 0, 0
7	4	2, 9, 17, 41	100, 237, 0, 0
8	4	1, 10, 13, 40	80, 0, 192, 0
9	4	0, 11, 12, 41	79, 0, 191, 0
10	4	5, 11, 12, 14	39, 0, 192, 0
11	4	4, 10, 13, 15	38, 255, 192, 0
12	3	0, 3, 22	221, 81, 0
13	3	1, 2, 23	221, 80, 0
14	3	0, 4, 24	154, 55, 0
15	3	1, 5, 25	154, 55, 0
16	4	3, 5, 6, 26	200, 130, 167, 0
17	4	2, 4, 7, 27	199, 129, 167, 0
18	4	0, 2, 8, 28	225, 18, 167, 0
19	4	1, 3, 9, 29	225, 18, 167, 0
20	4	1, 4, 12, 30	144, 63, 88, 0
21	4	0, 5, 13, 31	143, 63, 88, 0

Table 199—Base matrix G_0 " (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
22	4	2, 4, 14, 32	212, 70, 230, 0
23	4	3, 5, 15, 33	212, 70, 230, 0
24	4	0, 2, 11, 34	137, 232, 89, 0
25	4	1, 3, 10, 35	137, 232, 88, 0
26	4	0, 4, 7, 36	4, 94, 38, 0
27	4	1, 5, 6, 37	4, 94, 37, 0
28	4	1, 3, 8, 38	199, 206, 118, 0
29	4	0, 2, 9, 39	198, 205, 118, 0
30	4	1, 4, 16, 42	130, 106, 178, 0
31	4	0, 5, 17, 43	129, 106, 178, 0
32	3	2, 12, 44	238, 112, 0
33	3	3, 13, 45	238, 112, 0
34	3	5, 15, 46	222, 200, 0
35	3	4, 14, 47	221, 199, 0
36	4	0, 2, 18, 48	64, 23, 87, 0
37	4	1, 3, 19, 49	64, 23, 87, 0
38	3	4, 6, 50	123, 214, 0
39	3	5, 7, 51	123, 214, 0
40	4	2, 4, 20, 52	147, 254, 176, 0
41	4	3, 5, 21, 53	147, 254, 176, 0
42	3	1, 9, 54	7, 42, 0
43	3	0, 8, 55	6, 41, 0
44	4	2, 4, 10, 56	131, 146, 211, 0
45	4	3, 5, 11, 57	131, 146, 211, 0
46	3	1, 17, 58	3, 83, 0
47	3	0, 16, 59	2, 82, 0
48	3	2, 19, 60	31, 78, 0
49	3	3, 18, 61	31, 77, 0
50	3	5, 20, 62	60, 160, 0
51	3	4, 21, 63	59, 160, 0
52	4	0, 2, 12, 64	139, 152, 112, 0

Table 199—Base matrix G_0'' (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
53	4	1, 3, 13, 65	139, 152, 112, 0

Table 200 represents the matrix G_0''' with a lifting size of 256.

Table 200—Base matrix G_0'''

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
0	4	4, 10, 12, 13	38, 255, 192, 0
1	6	1, 2, 7, 9, 13, 14	90, 64, 163, 188, 0, 0
2	6	0, 5, 6, 12, 14, 15	175, 192, 215, 0, 0, 0
3	6	2, 5, 7, 9, 15, 16	27, 170, 106, 45, 0, 0
4	4	2, 9, 16, 17	100, 237, 0, 0
5	4	0, 11, 17, 18	79, 191, 0, 0
6	4	5, 11, 18, 19	39, 192, 0, 0
7	6	0, 3, 6, 8, 19, 20	90, 65, 163, 188, 0, 0
8	6	1, 4, 7, 11, 20, 21	176, 192, 216, 0, 0, 0
9	6	3, 4, 6, 8, 21, 22	28, 170, 106, 45, 0, 0
10	4	3, 8, 22, 23	101, 237, 0, 0
11	4	1, 10, 12, 23	80, 0, 192, 0
12	3	0, 3, 24	221, 81, 0
13	3	1, 2, 25	221, 80, 0
14	3	0, 4, 26	154, 55, 0
15	3	1, 5, 27	154, 55, 0
16	4	3, 5, 6, 28	200, 130, 167, 0
17	4	2, 4, 7, 29	199, 129, 167, 0
18	4	0, 2, 8, 30	225, 18, 167, 0
19	4	1, 3, 9, 31	225, 18, 167, 0
20	4	1, 4, 11, 32	144, 63, 88, 0
21	4	0, 5, 12, 33	143, 63, 88, 0
22	4	2, 4, 19, 34	212, 70, 230, 0
23	4	3, 5, 13, 35	212, 70, 230, 0
24	4	0, 2, 18, 36	137, 232, 89, 0
25	4	1, 3, 10, 37	137, 232, 88, 0

Table 200—Base matrix G_0''' (continued)

Row index	Row degree	Column positions of non-NULL elements in row	Shift numbers of non-NULL elements in row
26	4	0, 4, 7, 38	4, 94, 38, 0
27	4	1, 5, 6, 39	4, 94, 37, 0
28	4	1, 3, 8, 40	199, 206, 118, 0
29	4	0, 2, 9, 41	198, 205, 118, 0
30	4	1, 4, 22, 42	130, 106, 178, 0
31	4	0, 5, 16, 43	129, 106, 178, 0
32	3	2, 11, 44	238, 112, 0
33	3	3, 12, 45	238, 112, 0
34	3	5, 13, 46	222, 200, 0
35	3	4, 19, 47	221, 199, 0
36	4	0, 2, 21, 48	64, 23, 87, 0
37	4	1, 3, 15, 49	64, 23, 87, 0
38	3	4, 6, 50	123, 214, 0
39	3	5, 7, 51	123, 214, 0
40	4	2, 4, 20, 52	147, 254, 176, 0
41	4	3, 5, 14, 53	147, 254, 176, 0
42	3	1, 9, 54	7, 42, 0
43	3	0, 8, 55	6, 41, 0
44	4	2, 4, 10, 56	131, 146, 211, 0
45	4	3, 5, 18, 57	131, 146, 211, 0
46	3	1, 16, 58	3, 83, 0
47	3	0, 22, 59	2, 82, 0
48	3	2, 15, 60	31, 78, 0
49	3	3, 21, 61	31, 77, 0
50	3	5, 20, 62	60, 160, 0
51	3	4, 14, 63	59, 160, 0
52	4	0, 2, 11, 64	139, 152, 112, 0
53	4	1, 3, 12, 65	139, 152, 112, 0

9.2.6.4 Channel Interleaving

Channel interleaving applies to the Forward Primary Broadcast Control Channel, the Forward Secondary Broadcast Control Channel, the Forward Quick Paging Channel, the Forward Shared Control Channel, the

Forward Data Channel, the Forward Broadcast and Multicast Services Channel, the Forward Superposed Channel, the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel, the Reverse Code Division Multiple Access Data Channel, and the Reverse Orthogonal Frequency Division Multiple Access Data Channel.

Channel interleaving follows the convolutional or turbo encoding, and consists of a bit-demultiplexing operation followed by a bit permuting operation. Channel interleaving shall not be performed after LDPC encoding. The encoding operation shall be followed by the sequence repetition operation.

9.2.6.4.1 Bit demultiplexing

The output bits generated by the rate-1/3 convolutional encoder or rate-1/3 tail-biting convolutional encoder shall be reordered according to the following procedure:

- a) All of the convolutional encoder output bits shall be demultiplexed into three sequences denoted V_0 , V_1 , V_2 . The encoder output bits shall be sequentially distributed from the V_0 sequence to the V_2 sequence with the first bit going to the V_0 sequence, the second bit going to the V_1 sequence, the third to the V_2 sequence, and the fourth to the V_0 sequence, etc.
- b) The V_0 , V_1 , and V_2 sequences shall be ordered according to $V_0V_1V_2$. That is, the V_0 sequence shall be first, the V_1 sequence shall be second, and the V_2 sequence shall be last.

The output bits generated by the rate-1/5 turbo encoder shall be reordered according to the following procedure:

- All of the turbo encoder output data bits (i.e., the $5N_{\text{TURBO}}$ bits output in the first N_{TURBO} clock periods) shall be demultiplexed into five sequences denoted U , V_0 , V_1 , V'_0 , and V'_1 . The encoder output bits shall be sequentially distributed from the U sequence to the V'_1 sequence with the first encoder output bit going to the U sequence, the second to the V_0 sequence, the third to the V_1 sequence, the fourth to the V'_0 sequence, the fifth to the V'_1 sequence, the sixth to the U sequence, etc.
- The 18 tail bits numbered 0 through 17 (i.e., the 18 bits generated during the last six clock periods) shall be distributed as follows: Tail bits numbered 0, 3, 6, 9, 12, and 15 shall go to the U sequence; the tail bits numbered 1, 4, and 7 shall go to the V_0 sequence; the tail bits numbered 2, 5, and 8 shall go to the V_1 sequence; the tail bits numbered 10, 13, and 16 shall go to the V'_0 sequence; and the tail bits numbered 11, 14, and 17 shall go to the V'_1 sequence.

9.2.6.4.1.1 Bit permuting

9.2.6.4.1.1.1 Pruned Bit Reversal Interleaver

A Pruned Bit Reversal Interleaver (PBRI) shall be used in bit permuting for the rate-1/3 convolutional code, the rate-1/3 tail-biting convolutional code, and the rate-1/5 turbo code. See 9.2.5.2 for a description of the Pruned Bit Reversal Interleaver algorithm. The PBRI takes in a sequence of inputs and outputs that sequence in interleaved order. The i^{th} output of the PBRI of size M is equal to the j^{th} input, where $j = \text{PBRI}(i, M)$, where the function PBRI is as defined in 9.2.5.2 and M is the length of the input sequence.

9.2.6.4.1.1.2 Bit permuting for convolutional code

For the rate-1/3 convolutional code, the output sequence shall be the permuted $V_0/V_1/V_2$ sequence, which shall be generated according to the following procedure:

- a) Let sequence A be the V_0 sequence permuted by a PBRI, sequence B be the V_1 sequence permuted by a PBRI and sequence C be the V_2 sequence permuted by a PBRI.
- b) The permuted $V_0/V_1/V_2$ sequence shall be equal to sequence A followed by sequence B followed by sequence C .

9.2.6.4.1.1.3 Bit permuting for turbo code

The demultiplexed bits shall be permuted into three separate interleaved blocks with turbo coding. The three blocks shall consist of the permuted U sequence, the permuted V_0/V'_0 sequence and the permuted V_1/V'_1 sequence.

The permuted U block shall be equal to the U sequence permuted by a PBRI.

The permuted V_0/V'_0 sequence shall be generated according to the following procedure:

- Let sequence A be the V_0 sequence permuted by a PBRI and sequence B be the V'_0 sequence permuted by a PBRI.
- The permuted V_0/V'_0 sequence shall consist of alternate bits from sequence A and sequence B i.e., the $2i^{\text{th}}$ entry in the permuted V_0/V'_0 sequence shall be equal to the i^{th} entry in sequence A and the $(2i + 1)^{\text{th}}$ entry in the permuted V_0/V'_0 sequence shall be equal to the i^{th} entry in sequence B.

The permuted V_1/V'_1 sequence shall be generated according to the following procedure:

- Let sequence A be the V_1 sequence permuted by a PBRI and sequence B be the V'_1 sequence permuted by a PBRI.
- The permuted V_1/V'_1 sequence shall consist of alternate bits from sequence A and sequence B i.e., the $2i^{\text{th}}$ entry in the permuted V_1/V'_1 sequence shall be equal to the i^{th} entry in sequence A and the $(2i + 1)^{\text{th}}$ entry in the permuted V_1/V'_1 sequence shall be equal to the i^{th} entry in sequence B.

For all channels on the Reverse Link and all Forward Link channels other than the Forward Data Channel, the output sequence shall consist of the permuted U sequence followed by the permuted V_0/V'_0 sequence followed by the permuted V_1/V'_1 sequence.

If the Forward Data Channel is transmitted using extended transmissions or two-frame transmissions, then the output sequence shall consist of the permuted U sequence, followed by the permuted V_0/V'_0 sequence, followed by the permuted V_1/V'_1 sequence. The FTC MAC Protocol determines whether extended transmissions are used for a given packet.

If the Forward Data Channel is transmitted without using extended transmissions and two-frame transmissions are not used, then the output sequence depends on the packet size and `MaxRateOneFifthPacketSize`, `MaxRateOneThirdPacketSize`, and `MaxRateOneHalfPacketSize`, which are determined as follows:

If an 8-interlace HARQ structure is used when `DuplexingMode = FDD`, then

- `MaxRateOneFifthPacketSize` equals `MaxRateOneFifthPacketSizeEightInterlaceFDD`.
- `MaxRateOneThirdPacketSize` equals `MaxRateOneThirdPacketSizeEightInterlaceFDD`.
- `MaxRateOneHalfPacketSize` equals `MaxRateOneHalfPacketSizeEightInterlaceFDD`.

If a 6-interlace HARQ structure is used when `DuplexingMode = FDD`, then

- `MaxRateOneFifthPacketSize` equals `MaxRateOneFifthPacketSizeSixInterlaceFDD`.
- `MaxRateOneThirdPacketSize` equals `MaxRateOneThirdPacketSizeSixInterlaceFDD`.
- `MaxRateOneHalfPacketSize` equals `MaxRateOneHalfPacketSizeSixInterlaceFDD`.

If `DuplexingMode = TDD44`, then

- `MaxRateOneFifthPacketSize` equals `MaxRateOneFifthPacketSizeTDD44`.
- `MaxRateOneThirdPacketSize` equals `MaxRateOneThirdPacketSizeTDD44`.
- `MaxRateOneHalfPacketSize` equals `MaxRateOneHalfPacketSizeTDD44`.

If DuplexingMode = TDD63, then

- MaxRateOneFifthPacketSize equals MaxRateOneFifthPacketSizeTDD63.
- MaxRateOneThirdPacketSize equals MaxRateOneThirdPacketSizeTDD63.
- MaxRateOneHalfPacketSize equals MaxRateOneHalfPacketSizeTDD63.

The FTC MAC Protocol (see 8.6) determines which HARQ interlacing structure is used for transmitting the Forward Data Channel packet.

Configuration attributes of the FTC MAC Protocol (see 8.6) are as follows:

- MaxRateOneFifthPacketSizeEightInterlaceFDD
- MaxRateOneFifthPacketSizeSixInterlaceFDD
- MaxRateOneFifthPacketSizeTDD44
- MaxRateOneFifthPacketSizeTDD63
- MaxRateOneThirdPacketSizeEightInterlaceFDD
- MaxRateOneThirdPacketSizeSixInterlaceFDD
- MaxRateOneThirdPacketSizeTDD44
- MaxRateOneThirdPacketSizeTDD63
- MaxRateOneHalfPacketSizeEightInterlaceFDD
- MaxRateOneHalfPacketSizeSixInterlaceFDD
- MaxRateOneHalfPacketSizeTDD44
- MaxRateOneHalfPacketSizeTDD63

If $N_{\text{PACKET\_BITS}} \leq \text{MaxRateOneFifthPacketSize}$, the output sequence shall consist of the permuted U sequence followed by the permuted V_0/V'_0 sequence followed by the permuted V_1/V'_1 sequence.

If $\text{MaxRateOneFifthPacketSize} < N_{\text{PACKET\_BITS}} \leq \text{MaxRateOneThirdPacketSize}$, the output sequence shall consist of the permuted U sequence followed by the permuted V_0/V'_0 sequence. The permuted V_1/V'_1 sequence shall be discarded.

If $\text{MaxRateOneThirdPacketSize} \leq N_{\text{PACKET\_BITS}} < \text{MaxRateOneHalfPacketSize}$, the output sequence shall consist of the permuted U sequence followed by the first $(N_{\text{TURBO}} + 3)$ bits of the permuted V_0/V'_0 sequence. The remaining $(N_{\text{TURBO}} + 3)$ bits of the V_0/V'_0 sequence and the permuted V_1/V'_1 sequence shall be discarded.

If $\text{MaxRateOneHalfPacketSize} \leq N_{\text{PACKET\_BITS}}$, the output sequence shall consist of the permuted U sequence followed by the first $\lfloor (N_{\text{TURBO}} + 3)/2 \rfloor$ bits of the permuted V_0/V'_0 sequence. The remaining bits of the V_0/V'_0 sequence and the permuted V_1/V'_1 sequence shall be discarded.

9.2.6.5 Sequence repetition

Sequence repetition applies to the Forward Primary Broadcast Control Channel, the Forward Secondary Broadcast Control Channel, the Forward Quick Paging Channel, the Forward Shared Control Channel, the Forward Reverse Activity Bit Channel, the Forward Data Channel (except the Packet Data Control Assignment Message), the Forward Broadcast and Multicast Services Channel, the Forward Superposed Data Channel, the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel, the Reverse Code Division Multiple Access Data Channel, and the Reverse Orthogonal Frequency Division Multiple Access Data Channel.

Let $\{a_0, a_1, \dots, a_{N-1}\}$ be the sequence of bits at the output of the channel interleaver. This sequence of bits shall be repeated sequence-by-sequence as many times as are necessary to provide all of the bits needed for

the modulation procedure for that packet. For the Forward Data Channel the Forward Broadcast and Multicast Services Channel, the Forward Superposed Data Channel, the Reverse CDMA Data Channel, and the Reverse Orthogonal Frequency Division Multiple Access Data Channel, the packets are transmitted in a HARQ approach utilizing multiple HARQ transmissions. The sequence-repetition operation shall provide enough bits for all of these retransmissions.

9.2.6.5.1 Inverted sequence repetition

Inverted sequence repetition shall apply to the Packet Data Control Assignment Message hop-ports allocated to the Forward Data Channel only.

Let $\{a_0, a_1, \dots, a_{N-1}\}$ be the sequence of bits at the output of the channel interleaver. This sequence of bits shall first be reversed to yield $\{a_{N-1}, a_{N-2}, \dots, a_1, a_0\}$. The reversed sequence shall be repeated sequence-by-sequence as many times as are necessary to provide all of the bits needed for the modulation procedure for that packet.

9.2.6.6 Data scrambling

Data scrambling shall be done on a frame-by-frame basis using the common real scrambling algorithm. For each PHY frame in which a subpacket is transmitted on a Physical Layer channel, the encoded stream of bits transmitted in that PHY frame is scrambled using the common real scrambling algorithm with a seed specified for that PHY frame. The input seed to the common real scrambling algorithm shall be different for different Physical Layer channels, and shall be specified explicitly in the channel of interest.

The data scrambling operation shall be performed as follows: Let $\{y_0, y_1, y_2, \dots\}$ be the sequence of bits generated after the sequence repetition operation. Let $\{x_0, x_1, x_2, \dots\}$ be the sequence of bits used in the PHY frame of interest. The data scrambling operation shall comprise of generating a sequence $\{s_0, s_1, s_2, \dots\}$ using the common real scrambling algorithm (see 9.2.5.2) and flipping the bit x_i for each i if the corresponding $s_i = -1$.

9.2.6.7 Modulation

The Forward Data Channel, the Forward Broadcast and Multicast Services Channel, the Forward Superposed Data Channel, and Reverse Orthogonal Frequency Division Multiple Access Data Channel shall use QPSK, 8-PSK, 16-QAM, and 64-QAM modulation. The mappings that are used to generate the complex modulation output symbols with QPSK, 8-PSK, 16-QAM, and 64-QAM modulation are specified in 9.2.6.7.1 through 9.2.6.7.4.

The sequence of modulation symbols output from the modulator shall be equivalent to those generated by the following approach:

- a) Let $y(0,0), y(0,1), \dots$ be the infinite-length sequence of bits at the output of the scrambler corresponding to sub-packet 0, $y(1,0), y(1,1), \dots$ the infinite-length sequence of bits at the output of the scrambler for sub-packet 1 and so on. Let t be the total number of sub-packets. Initialize a set of t counters $j_0, j_1, \dots, j_{t-1}$, to 0. Counter j_m is a pointer to the next bit to be modulated for the m^{th} sub-packet, where, m is the index of the subpacket and takes values from 0 to $t-1$. Initialize another counter $k = 0$, which counts the total number of modulation symbols generated.
- b) Let q be the desired modulation order of the next modulation symbol and m be the desired subpacket, as specified in the description of the channel of interest. Collect the sequence of q bits $y(m, j_m), y(m, j_m + 1), \dots, y(m, j_m + q - 1)$ from the m^{th} subpacket and denote it as the sequence of bits $s_0, s_1, \dots, s_{q-1}$.
- c) The sequence $s_0, s_1, \dots, s_{q-1}$ is then mapped to a modulation symbol corresponding to the modulation format in that frame. The modulation formats are QPSK, 8-PSK, 16-QAM, and 64-

QAM. The mapping of $s_0, s_1, \dots, s_{q-1}$ to modulation symbols shall be as specified in 9.2.6.7.1, 9.2.6.7.2, 9.2.6.7.3, and 9.2.6.7.4, respectively.

- d) Increment counter j_m by q . Increment counter k by 1.
- e) Repeat steps b) through d) until the desired number of modulation symbols have been generated.

9.2.6.7.1 QPSK Modulation

In the case of QPSK modulation, a group of 2 input bits (s_0, s_1) is mapped into a complex modulation symbol ($m_I(k), m_Q(k)$), as specified in Table 201. Figure 74 shows the signal constellation of the QPSK modulator.

Table 201—QPSK modulation table

Modulator input bits		Modulation symbols	
s_1	s_0	$m_I(k)$	$m_Q(k)$
0	0	D	D
0	1	−D	D
1	0	D	−D
1	1	−D	−D

NOTE— $D = 1/(\sqrt{2})$.

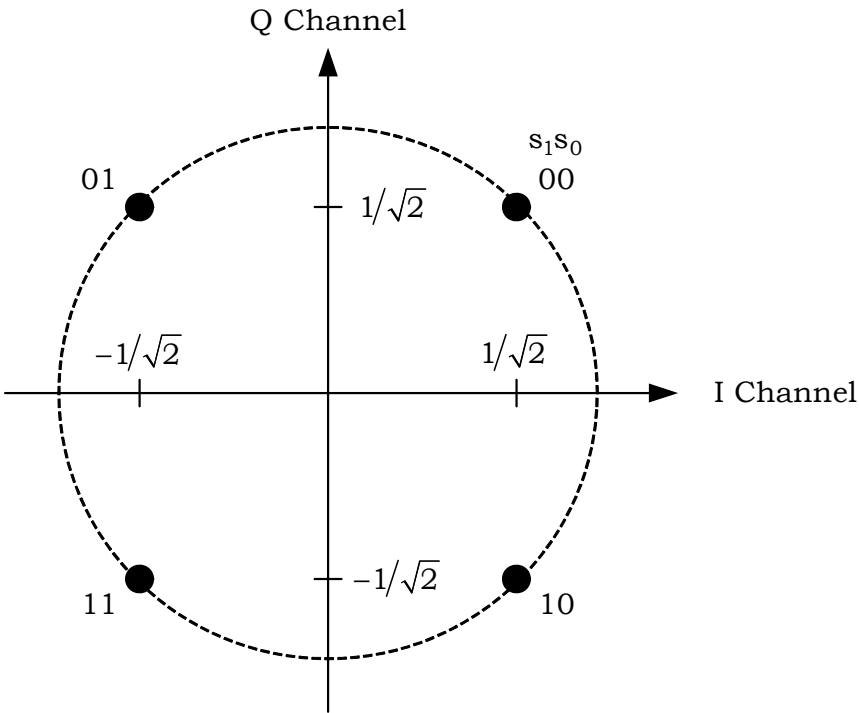


Figure 74—Signal constellation for QPSK modulation

9.2.6.7.2 8-PSK modulation

In the case of 8-PSK modulation, a group of 3 input bits (s_0, s_1, s_2) is mapped into a complex modulation symbol ($m_I(k), m_Q(k)$), as specified in Table 202. Figure 75 shows the signal constellation of the 8-PSK modulator.

Table 202—8-PSK modulation table

Modulator input bits			Modulation symbols	
s_2	s_1	s_0	$m_I(k)$	$m_Q(k)$
0	0	0	C	S
0	0	1	S	C
0	1	1	−S	C
0	1	0	−C	S
1	1	0	−C	−S
1	1	1	−S	−C
1	0	1	S	−C
1	0	0	C	−S

NOTE—C = $\cos(\pi/8)$ and S = $\sin(\pi/8)$

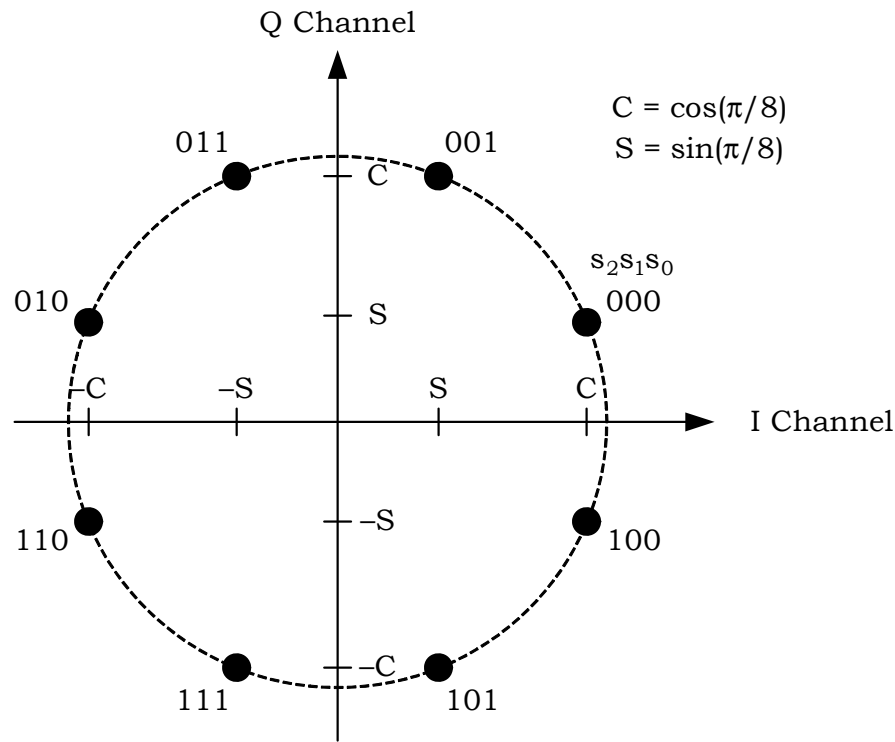


Figure 75—Signal constellation for 8-PSK modulation

9.2.6.7.3 16-QAM modulation

In the case of 16-QAM modulation, a group of 4 input bits (s_0, s_1, s_2, s_3) is mapped into a complex modulation symbol ($m_I(k), m_Q(k)$), as specified in Table 203. Figure 76 shows the signal constellation of the 16-QAM modulator.

Table 203—16-QAM modulation table

Modulator input bits				Modulation symbols	
s_3	s_2	s_1	s_0	$m_I(k)$	$m_Q(k)$
0	0	0	0	3A	3A
0	0	0	1	A	3A
0	0	1	1	−A	3A
0	0	1	0	−3A	3A
0	1	0	0	3A	A
0	1	0	1	A	A
0	1	1	1	−A	A
0	1	1	0	−3A	A
1	1	0	0	3A	−A
1	1	0	1	A	−A
1	1	1	1	−A	−A
1	1	1	0	−3A	−A
1	0	0	0	3A	−3A
1	0	0	1	A	−3A
1	0	1	1	−A	−3A
1	0	1	0	−3A	−3A

NOTE— $A = 1/(\sqrt{10})$

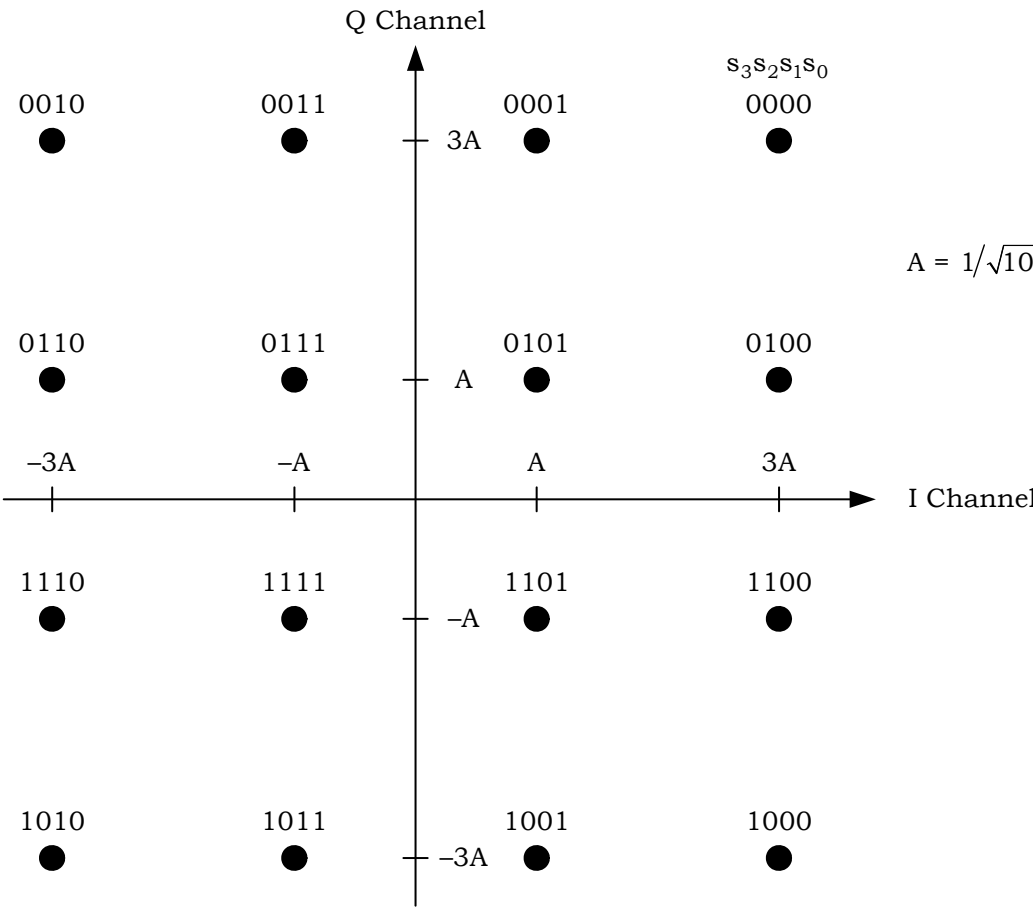


Figure 76—Signal constellation for 16-QAM modulation

9.2.6.7.4 64-QAM modulation

In the case of 64-QAM modulation, a group of 6 input bits ($s_0, s_1, s_2, s_3, s_4, s_5$) is mapped into a complex modulation symbol ($m_I(k), m_Q(k)$), as specified in Table 204. Figure 77 shows the signal constellation of the 64-QAM modulator.

Table 204—64 QAM modulation table

Modulator input bits						Modulation symbols	
s_5	s_4	s_3	s_2	s_1	s_0	$m_I(k)$	$m_Q(k)$
0	0	0	0	0	0	7A	7A
0	0	1	0	0	0	7A	5A
0	1	1	0	0	0	7A	3A
0	1	0	0	0	0	7A	A
1	1	0	0	0	0	7A	−A
1	1	1	0	0	0	7A	−3A

Table 204—64 QAM modulation table (continued)

Modulator input bits						Modulation symbols	
s_5	s_4	s_3	s_2	s_1	s_0	$m_I(k)$	$m_Q(k)$
1	0	1	0	0	0	7A	-5A
1	0	0	0	0	0	7A	-7A
0	0	0	0	0	1	5A	7A
0	0	1	0	0	1	5A	5A
0	1	1	0	0	1	5A	3A
0	1	0	0	0	1	5A	A
1	1	0	0	0	1	5A	-A
1	1	1	0	0	1	5A	-3A
1	0	1	0	0	1	5A	-5A
1	0	0	0	0	1	5A	-7A
0	0	0	0	1	1	3A	7A
0	0	1	0	1	1	3A	5A
0	1	1	0	1	1	3A	3A
0	1	0	0	1	1	3A	A
1	1	0	0	1	1	3A	-A
1	1	1	0	1	1	3A	-3A
1	0	1	0	1	1	3A	-5A
1	0	0	0	1	1	3A	-7A
0	0	0	0	1	0	A	7A
0	0	1	0	1	0	A	5A
0	1	1	0	1	0	A	3A
0	1	0	0	1	0	A	A
1	1	0	0	1	0	A	-A
1	1	1	0	1	0	A	-3A
1	0	1	0	1	0	A	-5A
1	0	0	0	1	0	A	-7A
0	0	0	1	1	0	-A	7A
0	0	1	1	1	0	-A	5A
0	1	1	1	1	0	-A	3A
0	1	0	1	1	0	-A	A
1	1	0	1	1	0	-A	-A
1	1	1	1	1	0	-A	-3A

Table 204—64 QAM modulation table (continued)

Modulator input bits						Modulation symbols	
s ₅	s ₄	s ₃	s ₂	s ₁	s ₀	m _I (k)	m _Q (k)
1	0	1	1	1	0	−A	−5A
1	0	0	1	1	0	−A	−7A
0	0	0	1	1	1	−3A	7A
0	0	1	1	1	1	−3A	5A
0	1	1	1	1	1	−3A	3A
0	1	0	1	1	1	−3A	A
1	1	0	1	1	1	−3A	−A
1	1	1	1	1	1	−3A	−3A
1	0	1	1	1	1	−3A	−5A
1	0	0	1	1	1	−3A	−7A
0	0	0	1	0	1	−5A	7A
0	0	1	1	0	1	−5A	5A
0	1	1	1	0	1	−5A	3A
0	1	0	1	0	1	−5A	A
1	1	0	1	0	1	−5A	−A
1	1	1	1	0	1	−5A	−3A
1	0	1	1	0	1	−5A	−5A
1	0	0	1	0	1	−5A	−7A
0	0	0	1	0	0	−7A	7A
0	0	1	1	0	0	−7A	5A
0	1	1	1	0	0	−7A	3A
0	1	0	1	0	0	−7A	A
1	1	0	1	0	0	−7A	−A
1	1	1	1	0	0	−7A	−3A
1	0	1	1	0	0	−7A	−5A
1	0	0	1	0	0	−7A	−7A

NOTE—A = $1/\sqrt{42}$

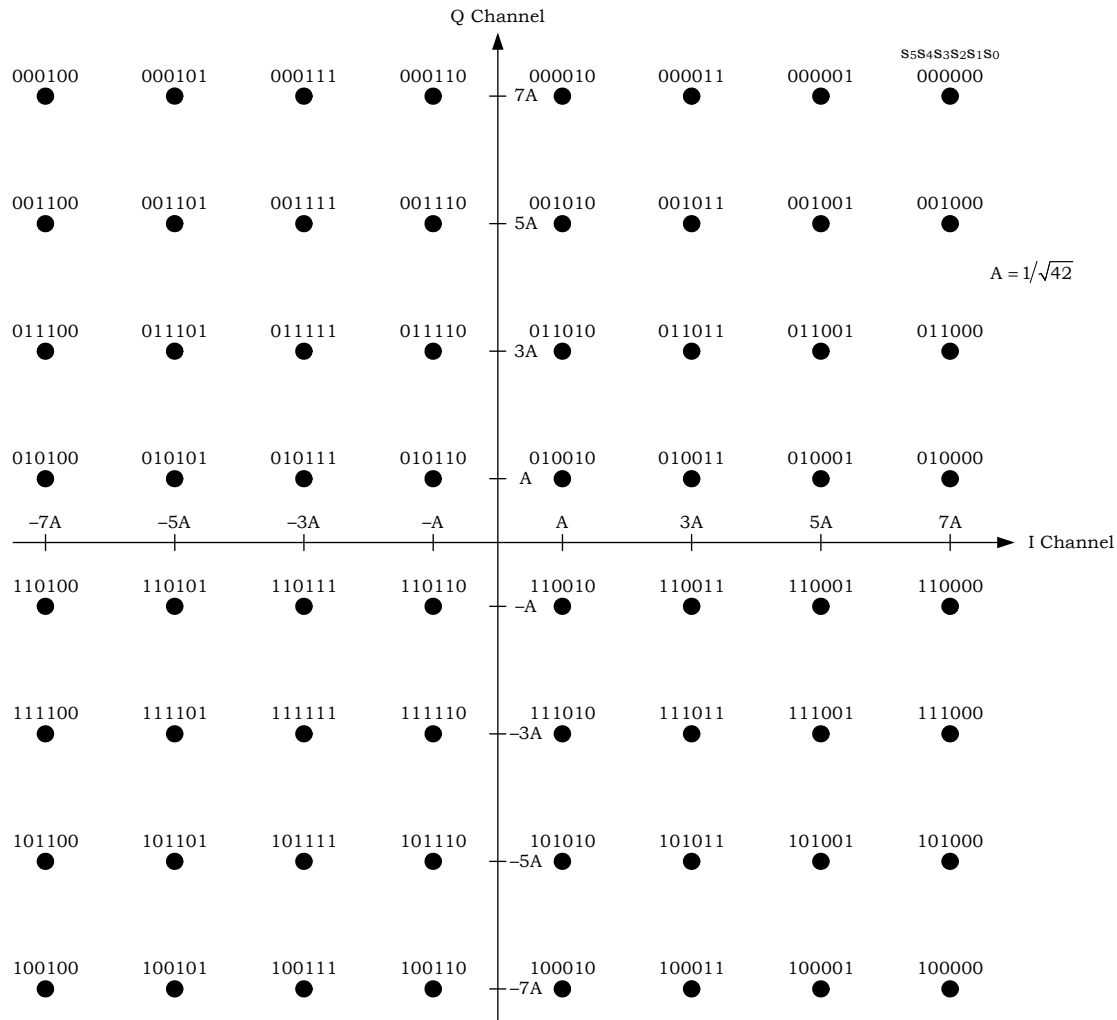


Figure 77—Signal constellation for 64-QAM modulation

9.2.6.7.5 Hierarchical modulation

In general, layered modulation can be a superposition of any two modulation schemes. For Broadcast and Multicast Services, a QPSK enhancement layer may be superposed on a base QPSK or 16-QAM layer to obtain the resultant signal constellation. The energy ratio r is the power ratio between the base layer and the enhancement layer. The enhancement layer is rotated by the angle θ in the counter-clockwise direction, which is specified by the field RotationalAngle in the BroadcastParameters Message, which is public data of the Broadcast Protocol Suite.

9.2.6.7.5.1 Modulation with QPSK base layer and QPSK enhancement layer

Each modulation symbol contains 4 bits, $[s_3, s_2, s_1, s_0]$, where s_2 and s_0 shall be from the base layer and s_3 and s_1 shall be from the enhancement layer. For the energy ratio r between the base and enhancement

layer, $\alpha = \sqrt{\frac{r}{2(1+r)}}$ and $\beta = \sqrt{\frac{1}{2(1+r)}}$ define the constellation completely.

Figure 78 shows the signal constellation of the layered modulator. The complex modulation symbol $S = (m_I, m_Q)$ for each $[s_3, s_2, s_1, s_0]$ is specified in Table 205.

Table 205—Layered modulation table with QPSK base layer and QPSK enhancement layer

Modulator input bits				Modulation symbols	
s_3	s_2	s_1	s_0	$m_I(k)$	$m_Q(k)$
0	0	0	0	$\alpha + \sqrt{2} \cos(\theta + \pi/4)\beta$	$\alpha + \sqrt{2} \sin(\theta + \pi/4)\beta$
0	0	1	0	$\alpha + \sqrt{2} \cos(\theta + 3\pi/4)\beta$	$\alpha + \sqrt{2} \sin(\theta + 3\pi/4)\beta$
1	0	0	0	$\alpha + \sqrt{2} \cos(\theta + 7\pi/4)\beta$	$\alpha + \sqrt{2} \sin(\theta + 7\pi/4)\beta$
1	0	1	0	$\alpha + \sqrt{2} \cos(\theta + 5\pi/4)\beta$	$\alpha + \sqrt{2} \sin(\theta + 5\pi/4)\beta$
0	0	1	1	$-\alpha + \sqrt{2} \cos(\theta + \pi/4)\beta$	$\alpha + \sqrt{2} \sin(\theta + \pi/4)\beta$
0	0	0	1	$-\alpha + \sqrt{2} \cos(\theta + 3\pi/4)\beta$	$\alpha + \sqrt{2} \sin(\theta + 3\pi/4)\beta$
1	0	1	1	$-\alpha + \sqrt{2} \cos(\theta + 7\pi/4)\beta$	$\alpha + \sqrt{2} \sin(\theta + 7\pi/4)\beta$
1	0	0	1	$-\alpha + \sqrt{2} \cos(\theta + 5\pi/4)\beta$	$\alpha + \sqrt{2} \sin(\theta + 5\pi/4)\beta$
1	1	0	0	$\alpha + \sqrt{2} \cos(\theta + \pi/4)\beta$	$-\alpha + \sqrt{2} \sin(\theta + \pi/4)\beta$
1	1	1	0	$\alpha + \sqrt{2} \cos(\theta + 3\pi/4)\beta$	$-\alpha + \sqrt{2} \sin(\theta + 3\pi/4)\beta$
0	1	0	0	$\alpha + \sqrt{2} \cos(\theta + 7\pi/4)\beta$	$-\alpha + \sqrt{2} \sin(\theta + 7\pi/4)\beta$
0	1	1	0	$\alpha + \sqrt{2} \cos(\theta + 5\pi/4)\beta$	$-\alpha + \sqrt{2} \sin(\theta + 5\pi/4)\beta$
1	1	1	1	$-\alpha + \sqrt{2} \cos(\theta + \pi/4)\beta$	$-\alpha + \sqrt{2} \sin(\theta + \pi/4)\beta$
1	1	0	1	$-\alpha + \sqrt{2} \cos(\theta + 3\pi/4)\beta$	$-\alpha + \sqrt{2} \sin(\theta + 3\pi/4)\beta$
0	1	1	1	$-\alpha + \sqrt{2} \cos(\theta + 7\pi/4)\beta$	$-\alpha + \sqrt{2} \sin(\theta + 7\pi/4)\beta$
0	1	0	1	$-\alpha + \sqrt{2} \cos(\theta + 5\pi/4)\beta$	$-\alpha + \sqrt{2} \sin(\theta + 5\pi/4)\beta$

NOTE— $\alpha = \sqrt{\frac{r}{2(1+r)}}$ and $\beta = \sqrt{\frac{1}{2(1+r)}}$, where r is the ratio of the base layer energy to the enhancement layer energy.

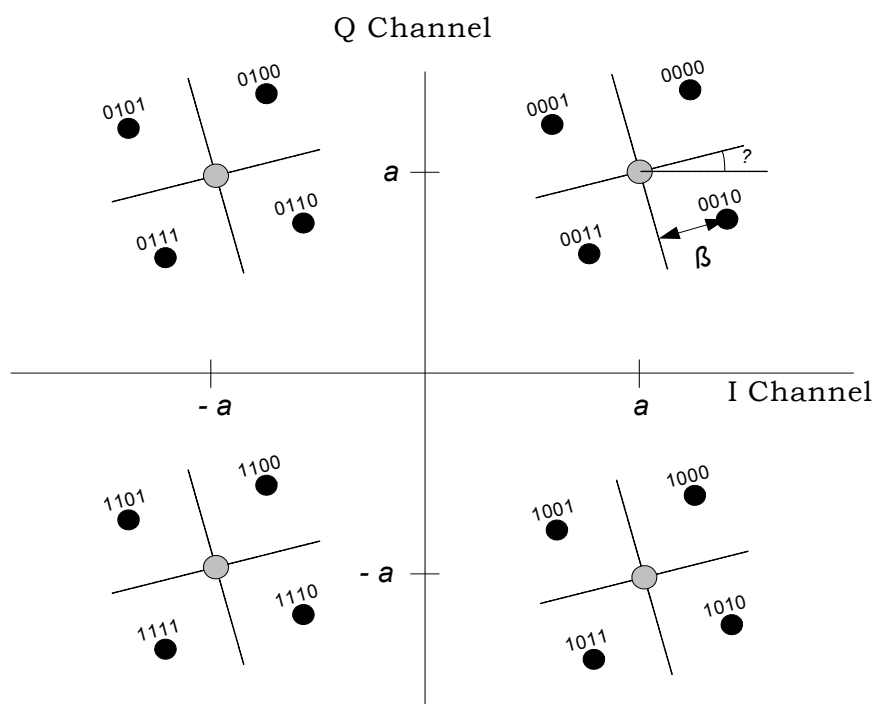


Figure 78—Signal constellation for layered modulation with a QPSK base layer and a QPSK enhancement layer

9.2.6.7.5.2 Modulation with 16-QAM base layer and QPSK enhancement layer

Each modulation symbol contains 6 bits, $[s_5, s_4, s_3, s_2, s_1, s_0]$, where s_4, s_3, s_1 and s_0 , shall be from the base layer and s_5 and s_2 , shall be from the enhancement layer. For the energy ratio r between the base layer and

enhancement layer, $\alpha = \sqrt{\frac{r}{10(1+r)}}$ and $\beta = \sqrt{\frac{1}{2(1+r)}}$ define the constellation completely.

Figure 79 shows the signal constellation of the layered modulator. The complex modulation symbol $S = (m_I, m_Q)$ for each $[s_5, s_4, s_3, s_2, s_1, s_0]$ is specified in Table 206.

Table 206—Layered modulation table with 16-QAM base layer and QPSK enhancement layer

Modulator input bits						Modulation symbols	
s_5	s_4	s_3	s_2	s_1	s_0	$m_I(k)$	$m_Q(k)$
0	0	0	0	0	0	$3\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
0	0	0	1	0	0	$3\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
1	0	0	0	0	0	$3\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
1	0	0	1	0	0	$3\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
0	0	0	1	0	1	$\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
0	0	0	0	0	1	$\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
1	0	0	1	0	1	$\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$

**Table 206—Layered modulation table with 16-QAM
base layer and QPSK enhancement layer (continued)**

Modulator input bits						Modulation symbols	
s ₅	s ₄	s ₃	s ₂	s ₁	s ₀	m _I (k)	m _Q (k)
1	0	0	0	0	1	$\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
0	0	0	1	1	0	$-3\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
0	0	0	0	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
1	0	0	1	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
1	0	0	0	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
0	0	0	0	1	1	$-\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
0	0	0	1	1	1	$-\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
1	0	0	0	1	1	$-\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
1	0	0	1	1	1	$-\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$3\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
1	0	1	0	0	0	$3\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
1	0	1	1	0	0	$3\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
0	0	1	0	0	0	$3\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
0	0	1	1	0	0	$3\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
1	0	1	1	0	1	$\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
1	0	1	0	0	1	$\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
0	0	1	1	0	1	$\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
0	0	1	0	0	1	$\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
1	0	1	1	1	0	$-3\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
1	0	1	0	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
0	0	1	1	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
0	0	1	0	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
1	0	1	0	1	1	$-\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
1	0	1	0	0	1	$-\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
0	0	1	0	1	1	$-\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
0	0	1	1	1	1	$-\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
1	1	0	0	0	0	$3\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
1	1	0	1	0	0	$3\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
0	1	0	0	0	0	$3\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
0	1	0	1	0	0	$3\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
1	1	0	1	0	1	$\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
1	1	0	0	0	1	$\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
0	1	0	1	0	1	$\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$

**Table 206—Layered modulation table with 16-QAM
base layer and QPSK enhancement layer (continued)**

Modulator input bits						Modulation symbols	
s ₅	s ₄	s ₃	s ₂	s ₁	s ₀	m _I (k)	m _Q (k)
0	1	0	0	0	1	$\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
1	1	0	1	1	0	$-3\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
1	1	0	0	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
0	1	0	1	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
0	1	0	0	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
1	1	0	0	1	1	$-\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
1	1	0	1	1	1	$-\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
0	1	0	0	1	1	$-\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
0	1	0	1	1	1	$-\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$-3\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
0	1	1	0	0	0	$3\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
0	1	1	1	0	0	$3\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
1	1	1	0	0	0	$3\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
1	1	1	1	0	0	$3\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
0	1	1	1	0	1	$\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
0	1	1	0	0	1	$\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
1	1	1	1	0	1	$\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
1	1	1	0	0	1	$\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
0	1	0	1	0	0	$-3\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
0	1	1	0	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
1	1	1	1	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
1	1	1	0	1	0	$-3\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$
0	1	1	0	1	1	$-\alpha + \sqrt{2}\cos(\theta + \pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + \pi/4)\beta$
0	1	1	1	1	1	$-\alpha + \sqrt{2}\cos(\theta + 3\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 3\pi/4)\beta$
1	1	1	0	1	1	$-\alpha + \sqrt{2}\cos(\theta + 7\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 7\pi/4)\beta$
1	1	1	1	1	1	$-\alpha + \sqrt{2}\cos(\theta + 5\pi/4)\beta$	$-\alpha + \sqrt{2}\sin(\theta + 5\pi/4)\beta$

NOTE— $\alpha = \sqrt{\frac{r}{10(1+r)}}$ and $\beta = \sqrt{\frac{1}{2(1+r)}}$, where r is the ratio of the base layer energy to the enhancement layer energy.

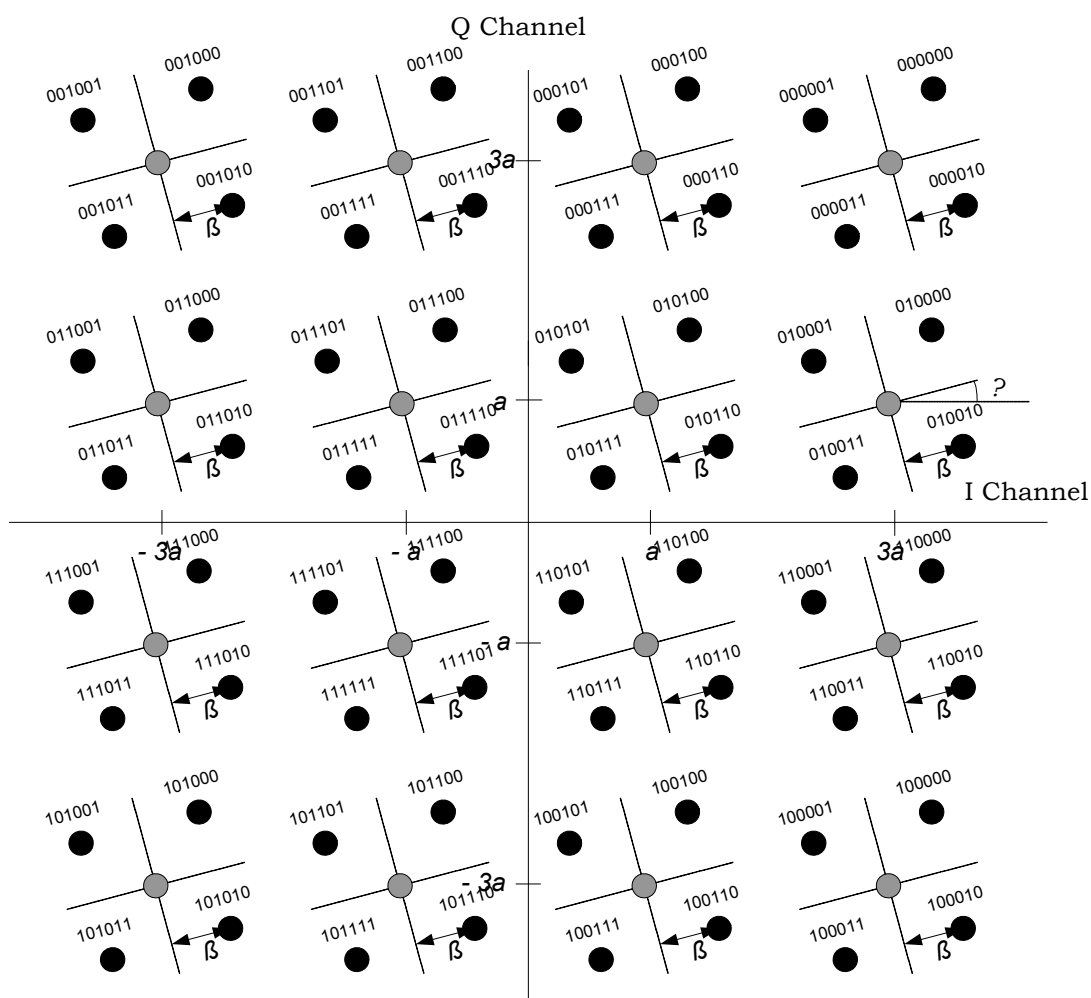


Figure 79—Signal constellation for layered modulation with a 16-QAM base layer and a QPSK enhancement layer

9.2.7 Orthogonal frequency division multiplexing structure and modulation parameters

9.2.7.1 Forward Link structure and modulation parameters

9.2.7.1.1 Superframe structure

Transmission on the Forward Link is divided into units of superframes. In FDD, each Forward Link superframe consists of a superframe preamble followed by a sequence of $N_{\text{FLPHYFrames}} = 25$ Forward Link PHY frames. If $\text{FDDHalfDuplexTDDPartition} = 1$ in $\text{DuplexingMode} = \text{FDD}$, then each of the Forward Link PHY frames is separated by a guard interval, whereas there is no separation when $\text{FDDHalfDuplexTDDPartition} = 0$ in $\text{DuplexingMode} = \text{FDD}$, where $\text{FDDHalfDuplexTDDPartition}$ is a field of the Overhead Messages Protocol.

Each superframe has an associated SuperframeIndex that is defined in 9.2.2. Both the superframe preamble and the Forward Link PHY frames further consist of a sequence of OFDM symbols. The structure of an

FDD Forward Link superframe is shown in Figure 80 for the case of $\text{FDDHalfDuplexTDDPartition} = 0$, and in Figure 81 for the case of $\text{FDDHalfDuplexTDDPartition} = 1$.

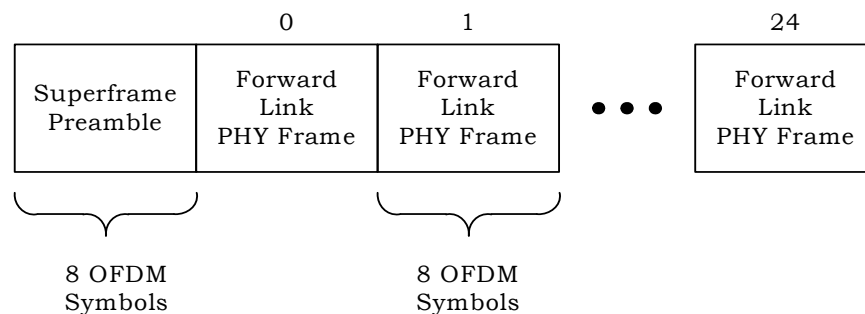


Figure 80—FDD Forward Link superframe structure if $\text{FDDHalfDuplexTDDPartition} = 0$

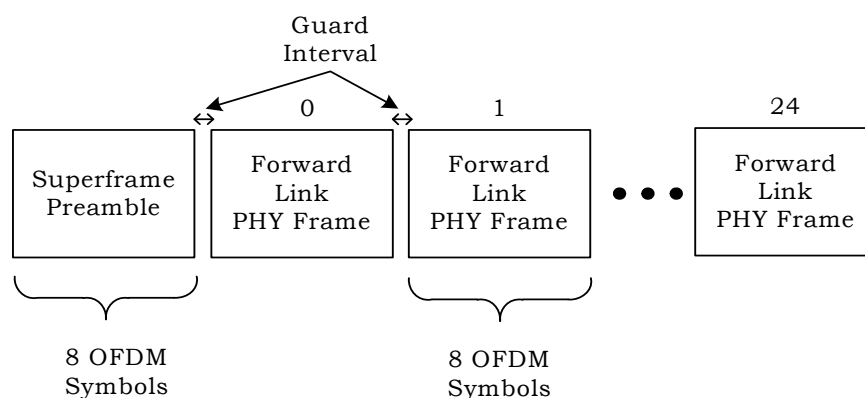


Figure 81—FDD Forward Link superframe structure if $\text{FDDHalfDuplexTDDPartition} = 1$

In TDD, each Forward Link superframe consists of a superframe preamble followed by a sequence of $N_{\text{FLPHYFrames}}$ Forward Link PHY frames. Each superframe has an associated SuperframeIndex that is incremented every superframe. If $\text{DuplexingMode} = \text{TDD44}$, there are $N_{\text{FLPHYFrames}} = 12$ FL PHY frames with $N_{\text{FLBurst}} = 4$ and $N_{\text{RLBurst}} = 4$; if $\text{DuplexingMode} = \text{TDD63}$, there are $N_{\text{FLPHYFrames}} = 18$ FL PHY frames with $N_{\text{FLBurst}} = 6$ and $N_{\text{RLBurst}} = 3$. The forward link superframe structures for TDD44 and TDD63 are shown in Figure 82 and Figure 83. Each FL PHY frame has an associated FL PHY frame index that is incremented every FL PHY frame.

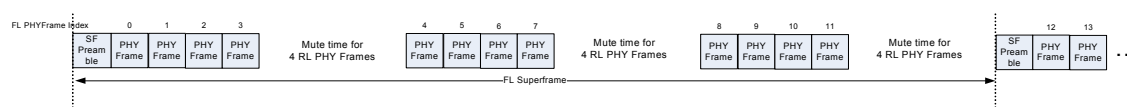


Figure 82—TDD44 Forward Link superframe structure

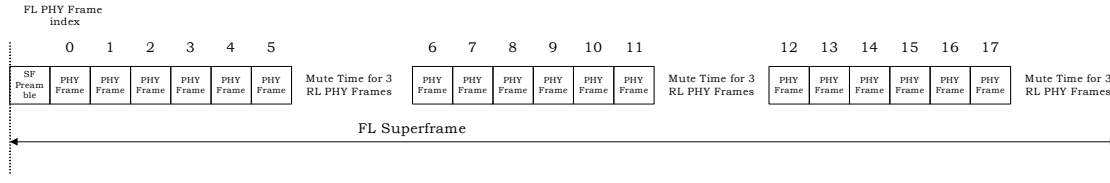


Figure 83—TDD63 Forward Link superframe structure

The superframe preamble consists of N_{PREAMBLE} OFDM symbols. Each FL PHY frame consists of $N_{\text{FRAME,F}}$ OFDM symbols. The mute time between each set of $N_{\text{FL\_BURST}}$ contiguous FL PHY frames in TDD is equal to the duration of $N_{\text{RL\_BURST}}$ contiguous RL PHY frames plus guard times $T_{\text{G,TDD,R}}$ and $T_{\text{G,TDD,F}}$. The total superframe duration is given by

$$T_{\text{SUPERFRAME}} = N_{\text{PREAMBLE}}T_s + N_{\text{FLPHYFrames}}N_{\text{FRAME,F}}T_s + N_{\text{RLPHYFrames}}N_{\text{FRAME,R}}T_s \\ + (T_{\text{G,TDD,F}} + T_{\text{G,TDD,R}}) \times (N_{\text{RLPHYFrames}}/N_{\text{RL\_BURST}})$$

where T_s denotes the OFDM symbol duration. Note that $N_{\text{FL\_BURST}}$, $N_{\text{RL\_BURST}}$, $N_{\text{FLPHYFrames}}$, and $N_{\text{RLPHYFrames}}$ are defined by parameters in AcqInfo block of the Connection Control Protocol. Formula used to compute the values of the parameters mentioned above is shown in 8.1.4.2.

The guard interval between a Forward Link PHY frames and the subsequent Reverse Link PHY frame is denoted as $T_{\text{g,TDD,F}}$, which shall be set to $N_{\text{FFT}}T_{\text{Chip}} \times 3/4$. The guard interval between a Reverse Link PHY frames and the subsequent Forward Link PHY frame is denoted as $T_{\text{g,TDD,R}}$, which shall be set to $N_{\text{FFT}}T_{\text{Chip}} \times 5/32$. Numerical values of the superframe parameters are listed in Table 208.

9.2.7.1.2 Orthogonal Frequency Division Multiplexing symbol structure

The Forward Link uses Orthogonal Frequency Division Multiplexing. As mentioned above, both the superframe preamble and the PHY frames consist of a sequence of OFDM symbols. An OFDM symbol consists of N_{FFT} individually modulated subcarriers that carry complex-valued data.

Complex-valued data are represented in the form $d = (d_{\text{re}}, d_{\text{im}})$, where d_{re} and d_{im} represent the real and imaginary components, respectively. The subcarriers in each OFDM symbol are indexed from 0 through $N_{\text{FFT}} - 1$. Here, N_{FFT} is given by the TotalNumSubcarriers parameter of the Overhead Messages Protocol.

The subcarriers indexed 0 through $N_{\text{GUARD, LEFT}} - 1$ as well as the subcarriers indexed $N_{\text{FFT}} - N_{\text{GUARD, RIGHT}}$ through $N_{\text{FFT}} - 1$ are designated as guard subcarriers and shall not be modulated [i.e., modulated with the complex value (0,0)], where, $N_{\text{GUARD, LEFT}} = N_{\text{GUARD}}/2$ and $N_{\text{GUARD, RIGHT}} = N_{\text{GUARD}}/2$, where N_{GUARD} is given by the NumGuardSubcarriers field of the Overhead Messages Protocol. Any subcarrier that is not a guard subcarrier is defined to be a usable subcarrier.

The OFDM symbol parameters shall be as specified in Table 207.

Table 207—Forward Link Orthogonal Frequency Division Multiplexing symbol numerology

Parameter	FFT Size			Units
	$N_{\text{FFT}} = 512$	$N_{\text{FFT}} = 1024$	$N_{\text{FFT}} = 2048$	
Chip Rate $1/T_{\text{CHIP}}$	4.9152	9.8304	19.6608	Mcps
Subcarrier Spacing $1/(T_{\text{CHIP}}N_{\text{FFT}})$	9.6	9.6	9.6	kHz
Bandwidth of Operation	2.5-5	5-10	10-20	MHz
Cyclic Prefix Duration $T_{\text{CP}} = N_{\text{CP}}N_{\text{FFT}}T_{\text{CHIP}}/16$ $N_{\text{CP}} = 1, 2, 3, \text{ or } 4$	6.51, 13.02, 19.53, or 26.04	6.51, 13.02, 19.53, or 26.04	6.51, 13.02, 19.53, or 26.04	μs
Windowing Guard Interval $T_{\text{WGI}} = N_{\text{FFT}}T_{\text{CHIP}}/32$	3.26	3.26	3.26	μs
Orthogonal Frequency Division Multiplexing Symbol Duration $T_s = N_{\text{FFT}}T_{\text{CHIP}}(1 + N_{\text{CP}}/16 + 1/32)$ $N_{\text{CP}} = 1, 2, 3, \text{ or } 4$	113.93, 120.44, 126.95, or 133.46	113.93, 120.44, 126.95, or 133.46	113.93, 120.44, 126.95, or 133.46	μs

The Forward Link supports two forms of Orthogonal Frequency Division Multiplexing data transmission—traditional Orthogonal Frequency Division Multiplexing and Rotational Orthogonal Frequency Division Multiplexing. The support of Rotational Orthogonal Frequency Division Multiplexing is optional at the access terminal and the access network.

Table 208—Forward Link Orthogonal Frequency Division Multiplexing superframe numerology

Parameter		Value	Units
N_{PREAMBLE} = Number of Orthogonal Frequency Division Multiplexing Symbols in the Superframe Preamble		8	
N_{FRAME} = Number of Orthogonal Frequency Division Multiplexing Symbols in a Forward Link PHY frame		8	
DuplexingMode = FDD	Number of FL PHY frames in a Superframe	25	
	Guard time between PHY frames when $\text{FDDHalfDuplexTDDPartition} = 1$ ($T_g = 3N_{\text{FFT}}T_{\text{CHIP}}/4$)	78.13	μs
	Superframe Duration ($T_{\text{SUPERFRAME}}$) when $\text{FDDHalfDuplexTDDPartition} = 0$ for $N_{\text{CP}} = 1, 2, 3, \text{ or } 4$	23.70, 25.05, 26.41, or 27.76	ms
	Superframe Duration ($T_{\text{SUPERFRAME}}$) when $\text{FDDHalfDuplexTDDPartition} = 1$ for $N_{\text{CP}} = 1, 2, 3, \text{ or } 4$	25.73, 27.08, 28.44, or 29.79	ms

Table 208—Forward Link Orthogonal Frequency Division Multiplexing superframe numerology (continued)

Parameter		Value	Units
DuplexingMode = TDD44	Number of FL PHY frames in a Superframe	12	
	Guard time between a FL PHY frame and the subsequent RL PHY frame ($T_{g,TDD,F} = 3N_{FFT}T_{CHIP}/4$)	78.13	μs
	Guard time between a RL PHY frame and the subsequent FL PHY frame ($T_{g,TDD,R} = 5N_{FFT}T_{CHIP}/32$)	16.28	μs
	Superframe Duration ($T_{SUPERFRAME}$) for $N_{CP} = 1, 2, 3, \text{ or } 4$	23.07, 24.37, 25.67, 26.98	ms
DuplexingMode = TDD63	Number of FL PHY frames in a Superframe	18	
	Guard time between a FL PHY frame and the subsequent RL PHY frame ($T_{g,TDD,F} = 3N_{FFT}T_{CHIP}/4$)	78.13	μs
	Guard time between a RL PHY frame and the subsequent FL PHY frame ($T_{g,TDD,R} = 5N_{FFT}T_{CHIP}/32$)	16.28	μs
	Superframe Duration ($T_{SUPERFRAME}$) for $N_{CP} = 1, 2, 3, \text{ or } 4$	25.80, 27.26, 28.72 or 30.18	μs

9.2.7.1.3 Orthogonal Frequency Division Multiplexing symbol start time

With respect to system time as defined in 9.2.2, the start time, $T_{START,SF}$, of the superframe with index SuperframeIndex with respect to the System Time is given by the product of SuperframeIndex with the superframe duration $T_{SUPERFRAME}$.

In the FDD mode, the start time of the k^{th} Orthogonal Frequency Division Multiplexing symbol in the superframe, k ranging from 0 to $N_{PREAMBLE} + N_{FRAME}N_{FLPHYFrames} - 1$, is given by $T_{START,SF} + kT_s + \lfloor k/N_{FRAME}T_g \rfloor$ where T_s is the OFDM symbol duration, T_g is the guard interval between two PHY frames when FDDHalfDuplexTDDPartition = 1. Otherwise, $T_g = 0$ when FDDHalfDuplexTDDPartition = 0.

In TDD, the start time of the k^{th} Orthogonal Frequency Division Multiplexing symbol in the superframe $T_{START,SF} + kT_s$, for $k < N_{\text{preamble}}$; or $T_{START,SF} + T_{PREAMBLE} + kT_s + \lfloor (k - N_{PREAMBLE}) / (N_{FRAME} \times N_{FLBurst}) \rfloor (T_{g,TDD,F} + T_{g,TDD,R} + N_{FRAME} \times T_s \times N_{RLBurst})$, for k ranging from $N_{PREAMBLE}$ to $N_{PREAMBLE} + N_{FRAME}N_{FLPHYFrames} - 1$, where $T_{PREAMBLE} = N_{PREAMBLE}T_s$.

9.2.7.1.4 Superframe preamble structure

Each superframe shall begin with a superframe preamble. The superframe preamble shall consist of $N_{PREAMBLE} = 8$ Orthogonal Frequency Division Multiplexing symbols that are indexed 0 through 7. The last three of these Orthogonal Frequency Division Multiplexing symbols (Orthogonal Frequency Division Multiplexing symbols 5 through 7) are TDM pilots, which are used for initial acquisition. These three Orthogonal Frequency Division Multiplexing symbols are denoted as TDM Pilot 1, TDM Pilot 2, and TDM Pilot 3, respectively. TDM Pilots 2 and 3 are additionally used to transmit the Other Sector Interference Channel as well. The TDM Pilot 1 Orthogonal Frequency Division Multiplexing symbol forms the Forward

Acquisition Channel, and the TDM Pilot 2 and 3 Orthogonal Frequency Division Multiplexing symbols form the Other Sector Interference Channel. The first Orthogonal Frequency Division Multiplexing symbol in the superframe preamble (i.e., the Orthogonal Frequency Division Multiplexing symbol with index 0) is used to transmit the Primary Broadcast Control Channel while the next four Orthogonal Frequency Division Multiplexing symbols (Orthogonal Frequency Division Multiplexing symbols indexed 1 through 4) are used to transmit the Secondary Broadcast Control Channel and the Quick Paging Channel in alternate superframes.

The structure of the superframe preamble is depicted in Figure 84. The different channels in the superframe preamble are described in 9.4.1.2.2.

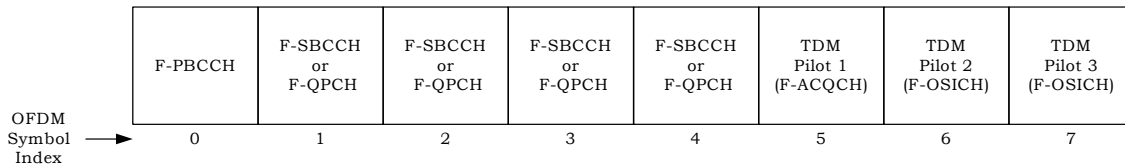


Figure 84—Superframe preamble structure

9.2.7.1.5 Reference Received Power Level and Reference Received Power Density

The Reference Received Power Level and Reference Received Power Density are defined for each sector. These quantities refer to the received power levels and the received power densities respectively. These quantities are used in other protocols; e.g., to compare the signal strengths of different sectors.

The received power of the Forward Primary Broadcast Control Channel is used as a Reference Received Power Level for the transmitting sector. The Reference Received Power Density per subcarrier of a sector is defined as P_{PBCCCH}/n_{sc} , where P_{PBCCCH} denotes the total received power of the Forward Primary Broadcast Control Channel and n_{sc} is the number of subcarriers carrying the Forward Primary Broadcast Control Channel in each OFDM symbol (note that this can change from superframe to superframe).

The PilotStrength of a given sector is defined as the ratio of the Reference Received Power Level of that sector to the total received power.

9.2.7.1.6 Reference Transmit Power

The power at which the first OFDM symbol of the preamble (the Forward Primary Broadcast Control Channel) is transmitted shall be defined as the Reference Transmit Power. This power may vary from sector to sector. For convenience of notation, the Reference Transmit Power of a given sector shall be set to 1 and the power levels and/or power densities of all other OFDM symbols transmitted by that sector shall be specified with respect to this power. Thus, if this standard states that a certain OFDM symbol is transmitted at power X , it should be inferred that the sector transmits that symbol at $XP_{REFERENCE}$, where $P_{REFERENCE}$ is the power of the first OFDM symbol in the preamble.

9.2.7.1.7 Forward Link interlace structure

The Forward Link PHY frames are divided into NumFLInterlaces interlaces, where NumFLInterlaces is equal to 8 if DuplexingMode = FDD, is equal to 4 if DuplexingMode = TDD44, and is equal to 6 if DuplexingMode = TDD63.

9.2.7.2 Reverse Link structure and modulation parameters

9.2.7.2.1 Superframe structure

Transmission on the Reverse Link is divided into units of superframes. In the FDD mode, each Reverse Link superframe consists of a sequence of $N_{\text{PHYFrames}} = 25$ Reverse Link PHY frames. Consecutive Reverse Link PHY frames are separated by a guard interval T_g in the FDD mode when $\text{FDDHalfDuplexTDDPartition} = 1$, whereas there is no separation when $\text{FDDHalfDuplexTDDPartition} = 0$, where, $\text{FDDHalfDuplexTDDPartition}$ is a field of the Overhead Messages Protocol.

Each superframe has an associated SuperframeIndex that is incremented every superframe. Each of the Reverse Link PHY frames consists of a sequence of Orthogonal Frequency Division Multiplexing symbols, where an Orthogonal Frequency Division Multiplexing symbol is as defined in 9.2.7.2.2. All but the Reverse Link PHY frame with index 0 in the FDD mode consists of $N_{\text{FRAME}} = 8$ Orthogonal Frequency Division Multiplexing symbols. If $\text{FDDHalfDuplexTDDPartition}$ is equal to 0 in $\text{DuplexingMode} = \text{FDD}$, the Reverse Link PHY frame with index 0 consists of $2N_{\text{FRAME}} = 16$ Orthogonal Frequency Division Multiplexing symbols, so as to cover the time occupied by the superframe preamble on the Forward Link (see 9.2.7.1.4 for the definition of the superframe preamble). If $\text{FDDHalfDuplexTDDPartition} = 1$, the Reverse Link PHY frame with index 0 in $\text{DuplexingMode} = \text{FDD}$ consists of only 8 Orthogonal Frequency Division Multiplexing symbols, which are aligned with the Forward Link PHY frame with index 0. The time corresponding to the Forward Link superframe preamble is left blank on the Reverse Link. The structure of an FDD Reverse Link superframe is shown in Figure 85 for $\text{FDDHalfDuplexTDDPartition} = 0$ and in Figure 86 for $\text{FDDHalfDuplexTDDPartition} = 1$.

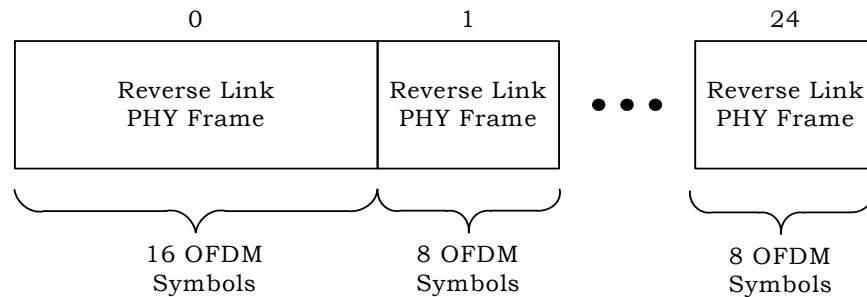


Figure 85—FDD Reverse Link superframe structure if $\text{FDDHalfDuplexTDDPartition} = 0$

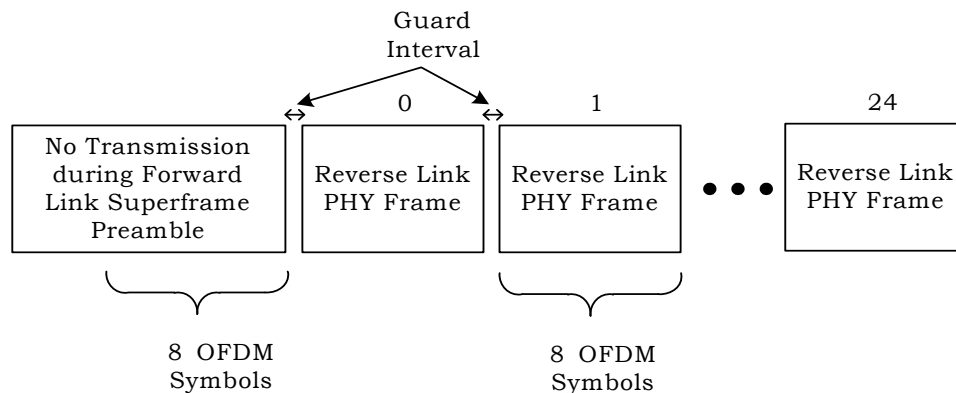


Figure 86—FDD Reverse Link superframe structure if $\text{FDDHalfDuplexTDDPartition} = 1$

In TDD, each Reverse Link superframe consists of a sequence of $N_{\text{RLPHYFrames}}$ Reverse Link PHY frames. Each superframe has an associated SuperframeIndex that is incremented every superframe. Each RL PHY frame has an associated RL PHY frame index that is incremented every RL PHY frame. If DuplexingMode = TDD44, there are $N_{\text{RLPHYFrames}} = 12$ RL PHY frames with $N_{\text{FL\_Burst}} = 4$ and $N_{\text{RL\_Burst}} = 4$; if DuplexingMode = TDD63, there are $N_{\text{RLPHYFrames}} = 9$ RL PHY frames with $N_{\text{FL\_Burst}} = 6$ and $N_{\text{RL\_Burst}} = 3$. The reverse link superframe structures for TDD44 and TDD63 are shown in Figure 87 and Figure 88.

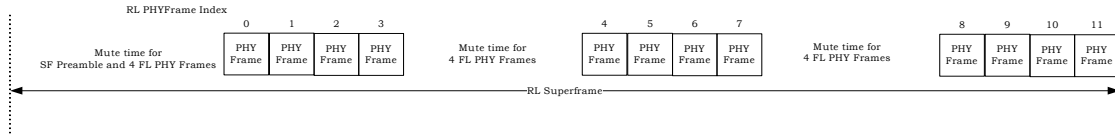


Figure 87—TDD44 Reverse Link superframe structure

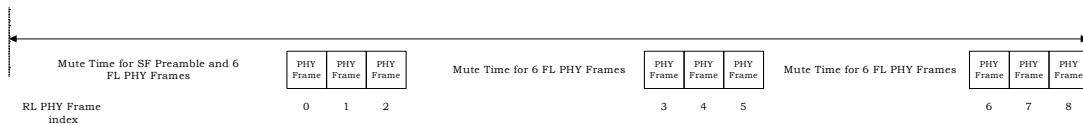


Figure 88—TDD63 Reverse Link superframe structure

Each RL PHY frame consists of N_{FRAME} OFDM symbols. N_{RLBurst} contiguous RL PHY frames forms a RL burst. The guard interval between a Forward Link PHY frames and the subsequent Reverse Link PHY frame is denoted as $T_{g,\text{TDD},F}$, which shall be set to $N_{\text{FFT}}T_{\text{Chip}} \times 3/4$. The guard interval between a Reverse Link PHY frames and the subsequent Forward Link PHY frame is denoted as $T_{g,\text{TDD},R}$, which shall be set to $N_{\text{FFT}}T_{\text{Chip}} \times 5/32$. Numerical values of the superframe parameters are listed in Table 208.

9.2.7.2.2 Orthogonal Frequency Division Multiplexing symbol structure

The Reverse Link uses Orthogonal Frequency Division Multiplexing. As mentioned previously, both the superframe preamble and the PHY frames consist of a sequence of Orthogonal Frequency Division Multiplexing symbols. An Orthogonal Frequency Division Multiplexing symbol consists of N_{FFT} individually modulated subcarriers that carry complex-valued data. Complex-valued data are represented in the form $d = (d_{\text{re}}, d_{\text{im}})$, where d_{re} and d_{im} represent the real and imaginary components, respectively. The subcarriers in each Orthogonal Frequency Division Multiplexing symbol are indexed from 0 through $N_{\text{FFT}} - 1$, where N_{FFT} is given by the TotalNumSubcarriers field of the Overhead Messages Protocol.

The subcarriers indexed 0 through $N_{\text{GUARD, LEFT}} - 1$ as well as the subcarriers indexed $N_{\text{FFT}} - N_{\text{GUARD, RIGHT}}$ through $N_{\text{FFT}} - 1$ are designated as guard subcarriers and shall not be modulated [i.e., modulated with the complex value (0,0)], where, $N_{\text{GUARD, LEFT}} = N_{\text{GUARD}}/2$ and $N_{\text{GUARD, RIGHT}} = N_{\text{GUARD}}/2$, where N_{GUARD} is given by the NumGuardSubcarriers field of the Overhead Messages Protocol. Any subcarrier that is not a guard subcarrier is defined to be a usable subcarrier.

Additionally, a silence interval may be used as described in 9.2.13. In this case, no energy shall be transmitted on subcarriers 0 through $N_{\text{GUARD,LEFT,EFFECTIVE}} - 1$, where $N_{\text{GUARD,LEFT,EFFECTIVE}}$ is defined in 9.2.13. In the case when a silence interval is not present $N_{\text{GUARD,LEFT,EFFECTIVE}} = N_{\text{GUARD, LEFT}}$. An additional parameter $N_{\text{GUARD, EFFECTIVE}}$ shall be defined where $N_{\text{GUARD, EFFECTIVE}} = N_{\text{GUARD,LEFT,EFFECTIVE}} + N_{\text{GUARD, RIGHT}}$.

The Orthogonal Frequency Division Multiplexing symbol parameters for different FFT sizes shall be as specified in Table 209.

Table 209—Reverse Link Orthogonal Frequency Division Multiplexing symbol numerology

Parameter	FFT Size			Units
	$N_{\text{FFT}} = 512$	$N_{\text{FFT}} = 1024$	$N_{\text{FFT}} = 2048$	
Chip Rate $1/T_{\text{CHIP}}$	4.9152	9.8304	19.6608	Mcps
Subcarrier Spacing $1/(T_{\text{CHIP}}N_{\text{FFT}})$	9.6	9.6	9.6	kHz
Bandwidth of Operation	2.5–5	5–10	10–20	MHz
Cyclic Prefix Duration $T_{\text{CP}} = N_{\text{CP}}N_{\text{FFT}}T_{\text{CHIP}}/16$ $N_{\text{CP}} = 1, 2, 3, \text{ or } 4$	6.51, 13.02, 19.53, or 26.04	6.51, 13.02, 19.53, or 26.04	6.51, 13.02, 19.53, or 26.04	μs
Windowing Guard Interval $T_{\text{WGI}} = N_{\text{FFT}}T_{\text{CHIP}}/3$	3.26	3.26	3.26	μs
Orthogonal Frequency Division Multiplexing Symbol Duration $T_{\text{s}} = N_{\text{FFT}}T_{\text{CHIP}}(1 + N_{\text{CP}}/16 + 1/32)$ $N_{\text{CP}} = 1, 2, 3, \text{ or } 4$	113.93, 120.44, 126.95, or 133.46	113.93, 120.44, 126.95, or 133.46	113.93, 120.44, 126.95, or 133.46	μs

Table 210—Reverse Link Orthogonal Frequency Division Multiplexing superframe numerology

Parameter		Value	Units
N_{FRAME} = Number of Orthogonal Frequency Division Multiplexing Symbols in a Reverse Link PHY frame (except the Reverse Link PHY frame with index 0 if FDDHalfDuplexTDDPartition = 0)		8	
$2N_{\text{FRAME}}$ = Number of Orthogonal Frequency Division Multiplexing Symbols in the Reverse Link PHY frame with index 0 if FDDHalfDuplexTDDPartition = 0		16	
DuplexingMode = FDD	Number of RL PHY frames in a Superframe	25	
	Guard time between PHY frames when FDDHalfDuplexTDDPartition = 1 ($T_{\text{g}} = 3N_{\text{FFT}}T_{\text{CHIP}}/4$)	78.13	μs
	Superframe Duration ($T_{\text{SUPERFRAME}}$) when FDDHalfDuplexTDDPartition = 0 for $N_{\text{CP}} = 1, 2, 3, \text{ or } 4$	23.70, 25.05, 26.41, or 27.76	ms
	Superframe Duration ($T_{\text{SUPERFRAME}}$) when FDDHalfDuplexTDDPartition = 1 for $N_{\text{CP}} = 1, 2, 3, \text{ or } 4$	25.65, 27, 28.4, 29.7	ms

Table 210—Reverse Link Orthogonal Frequency Division Multiplexing superframe numerology (continued)

Parameter		Value	Units
DuplexingMode = TDD44	Number of FL PHY frames in a Superframe	12	
	Guard time between a FL PHY frame and the subsequent RL PHY frame ($T_{g,TDD,F} = 3N_{FFT}T_{CHIP}/4$)	78.13	μs
	Guard time between a RL PHY frame and the subsequent FL PHY frame ($T_{g,TDD,R} = 5N_{FFT}T_{CHIP}/32$)	16.28	μs
	Superframe Duration ($T_{SUPERFRAME}$) for $N_{CP} = 1, 2, 3, \text{ or } 4$	23.07, 24.37, 25.67, 26.98	ms
DuplexingMode = TDD63	Number of FL PHY frames in a Superframe	9	
	Guard time between a FL PHY frame and the subsequent RL PHY frame ($T_{g,TDD,F} = 3N_{FFT}T_{CHIP}/4$)	78.13	μs
	Guard time between a RL PHY frame and the subsequent FL PHY frame ($T_{g,TDD,R} = 5N_{FFT}T_{CHIP}/32$)	16.28	μs
	Superframe Duration ($T_{SUPERFRAME}$) for $N_{CP} = 1, 2, 3, \text{ or } 4$	25.80, 27.26, 28.72 or 30.18	ms

9.2.7.2.3 Orthogonal Frequency Division Multiplexing Symbol start time

The start time, $T_{START,SF}$, of the superframe with index SuperframeIndex with respect to the access terminal reference is given by the product of SuperframeIndex with the superframe duration $T_{SUPERFRAME}$.

In the FDD mode, if FDDHalfDuplexTDDPartition is set to 0, the start time of the k^{th} OFDM symbol in the superframe, ($k = 0, 1, \dots, N_{PREAMBLE} + N_{FRAME}N_{RLPHYFrames} - 1$), is given by $T_{START,SF} + kT_s$, where T_s is the OFDM symbol duration.

In the FDD mode, if FDDHalfDuplexTDDPartition is set to 1, the start time of the k^{th} OFDM symbol in the superframe, ($k = 0, 1, \dots, N_{FRAME}N_{PHYFrames} - 1$), shall be given by $T_{START,SF} + (k + N_{PREAMBLE})T_s + (\lfloor k/N_{FRAME} \rfloor + 1)T_g$, where T_s is the OFDM symbol duration, T_g is the guard interval between two PHY frames when FDDHalfDuplexTDDPartition = 1. Otherwise, $T_g = 0$ when FDDHalfDuplexTDDPartition = 0.

For TDD, the start time of the k^{th} Orthogonal Frequency Division Multiplexing symbol in the superframe given by $T_{START,SF} + kT_s + (N_{PREAMBLE} + N_{FRAME} \times N_{FLBurst}) \times T_s + T_{g,TDD,F} + \lfloor k / (N_{FRAME} \times N_{RLBurst}) \rfloor \times (T_{g,TDD,F} + T_{g,TDD,R} + N_{FRAME} \times T_s \times N_{FLBurst})$ for k ranging from 0 to $N_{FRAME}N_{RLPHYFrames} - 1$.

9.2.7.2.4 Reverse Link interlace structure

The Reverse Link PHY frames are divided into NumRLInterlaces interlaces, where NumRLInterlaces is equal to 8 if DuplexingMode = FDD, is equal to 4 if DuplexingMode = TDD44, and is equal to 3 if DuplexingMode = TDD63.

9.2.7.3 Time-domain processing

The sequences of N_{FFT} complex modulation symbols per Orthogonal Frequency Division Multiplexing symbol shall be converted to a complex baseband waveform using an inverse Fourier transform operation, a windowing operation, and an overlap-and-add operation as specified in 9.2.7.3.1 through 9.2.7.3.3. This processing is illustrated in Figure 89.

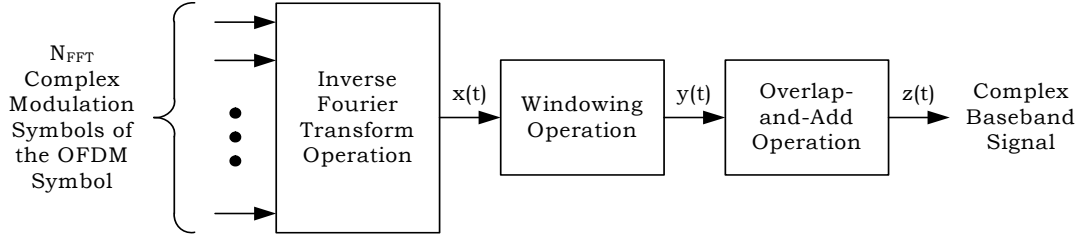


Figure 89—Time-domain processing

9.2.7.3.1 Inverse Fourier transform operation

Let X_k be the value of the complex modulation symbol on the k^{th} subcarrier of an Orthogonal Frequency Division Multiplexing symbol, where k is from 0 to $N_{\text{FFT}} - 1$. Then, the output of the inverse Fourier transform operation shall be given by Equation (21).

$$x(t) = \frac{1}{\sqrt{N_{\text{FFT}}}} \sum_{k=0}^{N_{\text{FFT}}-1} X_k e^{j2\pi(k - N_{\text{FFT}}/2)(t - T_{\text{CP}} - T_{\text{START}})/(N_{\text{FFT}}T_{\text{CHIP}})} \quad (21)$$

where T_{START} denotes the start time of the Orthogonal Frequency Division Multiplexing symbol (as specified in 9.3.1.4 and 9.4.1.4) and T_{CP} denotes the cyclic prefix durations for Orthogonal Frequency Division Multiplexing symbols in the superframe preamble and in PHY frames, respectively, and j denotes the complex number (0, 1).

9.2.7.3.2 Windowing Operation

The signal, $x(t)$, at the output of the inverse Fourier transform operation shall be multiplied by a window function, $w(t)$, giving a windowed signal of $y(t) = x(t)w(t)$. The window function shall be given by Equation (22).

$$w(t) = \begin{cases} 0 & , (t - T_{\text{START}}) < -T_{\text{WGI}} \\ 0.5 - 0.5 \cos\left(\frac{\pi(t + T_{\text{WGI}} - T_{\text{START}})}{T_{\text{WGI}}}\right) & , -T_{\text{WGI}} \leq (t - T_{\text{START}}) < 0 \\ 1 & , 0 \leq (t - T_{\text{START}}) < T_{\text{CP}} + T_{\text{FFT}} \\ 0.5 + 0.5 \cos\left(\frac{\pi(t - T_{\text{START}} - T_{\text{CP}} - T_{\text{FFT}})}{T_{\text{WGI}}}\right) & , T_{\text{CP}} + T_{\text{FFT}} \leq (t - T_{\text{START}}) < T_s \\ 0 & , (t - T_{\text{START}}) \geq T_s \end{cases} \quad (22)$$

where T_{START} denotes the start time of the Orthogonal Frequency Division Multiplexing symbol and T_s denotes the Orthogonal Frequency Division Multiplexing symbol duration, where $T_{\text{FFT}} = N_{\text{FFT}}T_{\text{CHIP}}$.

9.2.7.3.3 Overlap-and-Add operation

The windowed inverse-Fourier-transform output signals, $y(t)$, corresponding to all of the Orthogonal Frequency Division Multiplexing symbols shall be added together to create the final complex baseband waveform, $z(t)$. If DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 0, the neighboring Orthogonal Frequency Division Multiplexing symbols shall overlap for a duration of T_{WGI} , as illustrated in Figure 90. In all other cases, the neighboring OFDM symbols or Guard Times shall overlap for a duration of T_{WGI} , as illustrated in Figure 91.

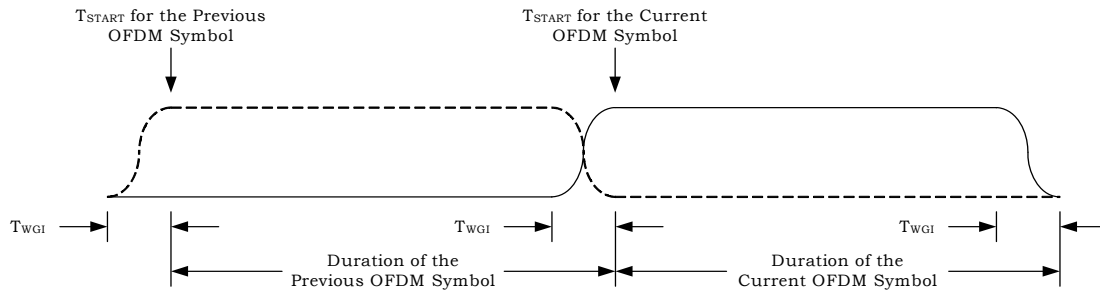


Figure 90—Overlap-and-Add operation for FDDHalfDuplexTDDPartition = 0

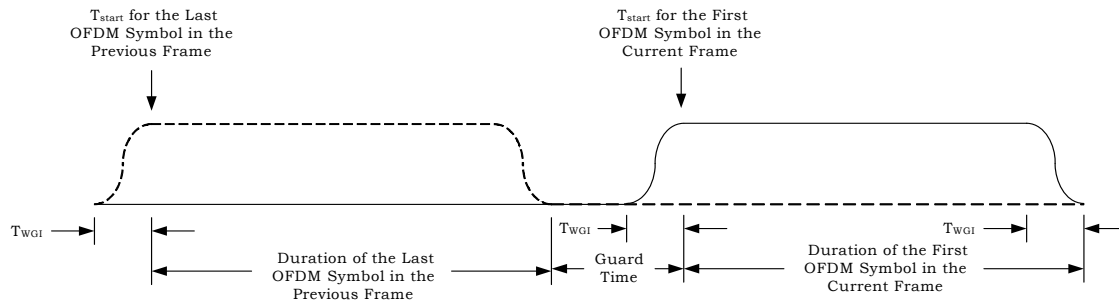


Figure 91—Overlap-and-Add operation for FDDHalfDuplexTDDPartition = 1

9.2.8 Multiple-input multiple-output procedures

The use of multiple transmit antennas for MIMO using both single-codeword and multi-codeword operations are described in 9.2.8.1. The operation of precoding is described in 9.2.8.2. The permutations matrices used for multi-codeword MIMO is described in 9.2.8.3. If the hop-permutation maps multiple hop-ports to the same subcarrier, the Physical Layer also supports the superposition of multiple (up to 4) waveforms on the same set of subcarriers, potentially using different precoding matrices. This is known as Space Division Multiple Access (SDMA). SDMA operation is described in 9.2.8.4.

9.2.8.1 Multiple transmit antennas

Multiple transmit antennas may be present at the sector transmitter. A logical antenna is a linear combination of physical antennas that is slowly varying over time and frequency, so that it appears to the receiver as a single physical antenna over a linear, time-invariant channel. The mapping between logical and

physical transmit antennas is beyond the scope of this standard. Note that transmission on a single logical antenna may involve transmission on any or all of the physical antennas.²⁵

Certain specific logical antennas are indexed from 0 to NumEffectiveAntennas – 1, where NumEffectiveAntennas is a parameter of the Overhead Messages Protocol. These specific logical antennas are referred to as “effective antennas.” Each effective antenna is associated with the Forward Common Pilot Channel and the Forward Channel Quality Indicator Pilot Channel, which are transmitted on that effective antenna. The radio channel response associated with each effective antenna may be estimated by the receiver using the corresponding Forward Common Pilot Channel or the corresponding Forward Channel Quality Indicator Pilot Channel signal. The effective antennas in the system are indexed 0 through NumEffectiveAntennas – 1. Operations such as time-domain processing and hop-sequence generation, described in 9.2.7.3 and 9.2.14 respectively, are identical for each of the effective antennas.

Transmission of some of the Physical Layer channels (e.g., the Forward Common Pilot Channel and the Forward Channel Quality Indicator Pilot Channel) is specified separately for each effective antenna. Some other Physical Layer channels (e.g., the Forward Data Channel when ResourceChannelMultiplexingMode = 1) are transmitted on effective antennas, or linear combinations of effective antennas that are known to the receiver. Some of the other channels (such as the Forward Preamble Pilot Channel) are transmitted on logical antennas that may be unrelated to the effective antennas.

A tile-antenna is a linear combination of the physical antennas present in the system, where the choice of linear combination is fixed over each tile, but may vary arbitrarily from tile to tile. The exact choice of the linear combinations used to construct tile-antennas is beyond the scope of this standard. Some of the Physical Layer channels (for example the Forward Dedicated Pilot Channel and the Forward Data Channel transmitted in the BRCH zone, when ChannelResourceMultiplexingMode = 2) are transmitted on tile-antennas.

Table 211 gives a summary of the Forward Link channels and the antennas used to transmit each of those channels. The exact choice of the antennas and transmission procedures of each of these channels are specified in the corresponding subclauses.

Table 211—Antennas used for various channels

Channel name	Antenna
Channels in the superframe preamble	A logical antenna (i.e., linear combination of physical antennas) that may or may not be related to effective antennas.
Channels in the Forward Link Control Segment	Tile antenna 0 when UseDRCHForFLCS = 0 and ResourceChannelMultiplexingMode = 2. Effective antenna 0, otherwise.
Forward Common Pilot Channel	Effective antennas 0 through NumEffectiveAntennas – 1. The i^{th} Forward Common Pilot Channel, if present, is transmitted on effective antenna i .
Forward Channel Quality Indicator Pilot Channel	Effective antennas 0 through NumEffectiveAntennas – 1. The i^{th} Forward Channel Quality Indicator Pilot Channel, if present, is transmitted on effective antenna i .
Forward Dedicated Pilot Channel	Tile antennas.

²⁵ An effective antenna may be constructed, for example, by using delay diversity on a set of physical antennas.

Table 211—Antennas used for various channels (continued)

Channel name	Antenna
Forward Beacon Pilot Channel	A logical antenna (i.e., linear combination of physical antennas) that may or may not be related to effective antennas or the superframe preamble.
Forward Data Channel	The Forward Data Channel antennas, where the Forward Data Channel antenna with index k is: 1. Tile-antenna k if both of the following criteria are satisfied: a. The resource being modulated is a BRCH resource. b. ResourceChannelMuxMode = 2. 2. Effective antenna k if both of the following conditions are satisfied: a. The resource being modulated is either a DRCH resource or ResourceChannelMuxMode = 1. b. No precoding is used. 3. A logical antenna that is a linear combination (known or signaled to the access terminal) of effective antennas and is as described in 9.2.8.1, if both of the following conditions are satisfied: a. The resource being modulated is either a DRCH resource or ResourceChannelMuxMode = 1. b. Precoding is used.
Forward Broadcast and Multicast Pilot Channel	A logical antenna (i.e., linear combination of physical antennas) that may or may not be related to effective antennas or the Forward Data Channel antennas.
Forward Broadcast and Multicast Services Channel	A logical antenna (i.e., linear combination of physical antennas) that may or may not be related to effective antennas or the Forward Data Channel antennas.
Forward Superposed Dedicated Pilot Channel	Tile antennas.
Forward Superposed Channel Quality Indicator Pilot Channel	Tile antennas.
Forward Superposed Data Channel	Tile antennas.

Here UseDRCHForFLCS, ResourceChannelMuxMode, and NumEffectiveAntennas are parameters of the Overhead Messages Protocol. DRCH and BRCH resources are described in 9.2.14.2 and precoding is described in 9.2.8.2.

9.2.8.2 Precoding

Precoding is a linear pre-processing that enables transmit beamforming for each Multiple Input Multiple Output layer. Closed-loop precoding is performed based on the feedback from the access terminals. Each such mapping is characterized by a particular precoding matrix.

A set of precoding matrices forms a codebook, from which an access terminal may feed back a preferred matrix to the access network through the logical channel r-bfch (reverse beamforming channel) as specified by the RCC MAC Protocol (see 8.7).

9.2.8.2.1 Use of precoding matrices

The columns of the precoding matrix define a set of spatial beams that can be used by the access network. Each precoding matrix is a matrix with NumEffectiveAntennas rows and SpatialOrder columns, which defines a mapping between the NumEffectiveAntennas effective antennas and N_{LAYERS} logical antennas that are used to transmit modulation symbols when precoding is used, where SpatialOrder and N_{LAYERS} (the number of layers used in the transmission) are specified by the FTC MAC Protocol (8.6), and NumEffectiveAntennas is a field of the Overhead Messages Protocol. If SISO transmission is used, N_{LAYERS} shall be set to 1.

The mapping between effective antennas and logical antennas used for precoding with precoding matrix W [specified by the FTC MAC Protocol (8.6)] shall be as follows:

- If the Readymade Codebook is used, let W^0 through $W^{N_{\text{LAYER}}-1}$ be the first N_{LAYERS} columns of the precoding matrix.
- If the Knockdown Codebook is used, let W^0 through $W^{N_{\text{LAYER}}-1}$ be any N_{LAYERS} columns of the precoding matrix. The choice of columns is specified by the FTC MAC Protocol (8.6).
- Let $N_{\text{TX-PREC-ANT}}$ be equal to MaxPrecodedTransmitAntennas, which is an attribute of the FTC MAC Protocol (8.6).
- The mapping between the effective antennas and the logical antenna with index k shall be given by the first $N_{\text{TX-PREC-ANT}}$ elements of W^k . In other words, in order to transmit a modulation symbol s on logical antenna k , the sector transmits $W^k(i) \times s$ on effective antenna i , where $0 \leq i < N_{\text{TX-PREC-ANT}}$, where $W^k(i)$ is the i^{th} element of the vector W^k .

9.2.8.2.2 Codebook types

Two classes of precoding codebooks are supported—knockdown codebook and readymade codebook.

9.2.8.2.2.1 Knockdown codebook

A knockdown precoder is constructed by assembling vectors chosen from a selected universal matrix. Each knockdown precoding codebook defines up to two universal unitary matrices. The access terminal selects one of the universal matrices as a preferred matrix, which is indicated by the preferred matrix index (PMI).

The access terminal also selects preferred column vectors from the preferred matrix. The selected column vectors are indicated by the vector bitmap (VBM). VBM and PMI are indicated in the RCC MAC Protocol (8.7).

If a codebook defines G $M_T \times M_T$ universal matrices, where M_T is the number of physical antennas and $G \leq 2$, the number of precoders for a transmission of rank M is $G \times C_M^M$, and the total number of precoding matrices is $G(2^{M_T} - 1)$.

9.2.8.2.2.1.1 Default knockdown codebooks

Two kinds of default knockdown codebooks have been predefined—the Binary unitary codebook and the Fourier matrix codebook.

9.2.8.2.2.1.1.1 Binary unitary codebook

This codebook is defined by the 4×4 identity matrix I_4 .

9.2.8.2.2.1.1.2 Fourier matrix based codebook

This consists of the $M \times M$ matrices $H^{(g)}$, where $H^{(g)}$ are defined by Equation (23).

$$H^{(g)} = \begin{bmatrix} h_{nm}^{(g)} \end{bmatrix} = \begin{bmatrix} e^{\left\{ j \frac{2\pi n}{M} \left(m + \frac{g}{G} \right) \right\}} \end{bmatrix} \quad (23)$$

where $m, n = 0, 1, \dots, M-1$.

As an example, for $M = 4$, and $G = 2$, the matrices are as follows:

$$H^{(0)} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix}, \quad H^{(1)} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ \frac{1+j}{\sqrt{2}} & \frac{-1+j}{\sqrt{2}} & \frac{-1-j}{\sqrt{2}} & \frac{1-j}{\sqrt{2}} \\ j & -j & j & -j \\ \frac{-1+j}{\sqrt{2}} & \frac{1+j}{\sqrt{2}} & \frac{1-j}{\sqrt{2}} & \frac{-1-j}{\sqrt{2}} \end{pmatrix}$$

9.2.8.2.2.2 Readymade codebook

An actual precoder is constructed by the first r vectors in a selected matrix where r is the required rank. A readymade precoding codebook defines up to 64 precoding matrices. The access terminal selects one of the precoding matrices as a preferred precoding matrix, which is indicated by the precoder index (PCI).

The readymade precoding codebooks are configurable through precoder codebook download mechanism. The rank is indicated either by the Channel Quality Indicator feedback in an implicit manner for Multi-Codeword Multiple Input Multiple Output or by the rank feedback in an explicit manner for Single Codeword Multiple Input Multiple Output. If a codebook defines K precoding matrices, the total number of precoders is K .

9.2.8.2.3 Downloadable codebook

Precoding codebooks, other than default codebooks, are configurable through codebook download. Downloadable codebooks are mandatory for the access terminals that support precoding. Both knockdown and readymade precoding codebooks are downloadable.

Some of the precoding matrices in a codebook may be grouped into clusters. In this case, matrices in a single cluster typically span only part of the space. The columns of the matrices in different clusters are used to form spatial beams covering spatially distinct groups of users. If the access terminal feeds back a beam index within a cluster, the access network treats this as an indication that it may schedule other access terminals on different clusters, i.e., allowing for Space Division Multiple Access (SDMA). However, the codebook may be formed by only a set of precoding matrices such that each spans the whole space. In this case, this codebook is used for precoding and is not intended to be used for SDMA.

The downloaded codebooks shall specify precoding matrices of size $\text{NumAntenna} \times \text{SpatialOrder}$. The precoding matrices used shall be downloaded from the FTC MAC Protocol (8.6), corresponding to the codebook specified by CodeBookID.

9.2.8.2.4 Random orthonormal ensemble

If ResourceChannelMuxMode = 2 and NumDRCHSubzones = 0, the access terminal should use a random orthonormal ensemble to compute the channel quality indicator value corresponding to a non-coded transmission. A random orthonormal ensemble with parameters (n, m) consists of a large collection of randomly generated unitary matrices of size $n \times m$. The following rule [Equation (24)] may be used to generate a matrix $H_{n \times m}$ from the random orthonormal ensemble:

$$H_{n \times m} = \frac{1}{\sqrt{n}} \begin{pmatrix} e^{2\pi j \varphi_0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & e^{2\pi j \varphi_{n-1}} \end{pmatrix} \begin{pmatrix} F_{00} & \cdots & F_{0,m-1} \\ \vdots & \ddots & \vdots \\ F_{n-1,0} & \cdots & F_{n-1,m-1} \end{pmatrix} \quad (24)$$

where

$$F_{ab} = e^{2\pi j ab/n} \text{ for } a = 0, 1, \dots, n-1$$

$$b = 0, 1, \dots, m-1$$

φ_0 through φ_{n-1} are random numbers drawn from the uniform distribution on $[0, 1]$

9.2.8.3 Permutation matrices for Multi-Codeword Multiple Input Multiple Output

Permutation matrices are used for data transmission in Multi-Codeword Multiple Input Multiple Output (see 9.4.1.2.5.7). There are $N!$ permutation matrices of order N that are numbered from 0 to $N! - 1$. All the permutation matrices up to order 4 are listed in 9.2.8.3.1 to 9.2.8.3.4.

9.2.8.3.1 Permutation matrices of order 1

There is only one matrix as follows: $P_0^1 = [1]$.

9.2.8.3.2 Permutation matrices of order 2

There are two matrices. They are enumerated as follows:

$$P_0^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, P_1^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

9.2.8.3.3 Permutation matrices of order 3

There are six matrices. They are enumerated as follows:

$$P_0^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, P_1^3 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, P_2^3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

$$P_3^3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, P_4^3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, P_5^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

9.2.8.3.4 Permutation matrices of order 4

There are twenty-four matrices. They are enumerated as follows:

$$\begin{aligned}
 P_0^4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, P_1^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, P_2^4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_3^4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\
 P_4^4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, P_5^4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_6^4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_7^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \\
 P_8^4 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_9^4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, P_{10}^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_{11}^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 P_{12}^4 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_{13}^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_{14}^4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, P_{15}^4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 P_{16}^4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_{17}^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_{18}^4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, P_{19}^4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 P_{20}^4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_{21}^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, P_{22}^4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_{23}^4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.
 \end{aligned}$$

9.2.8.4 SDMA operation

When the SDMA is used, the access network transmits different symbols to different access terminals through the same subcarrier. SDMA operation is summarized as follows:

The difference between channel quality in the absence and presence of SDMA is reflected by a differential channel quality that is fed back through r-bfch, which is a logical channel defined by the RCC MAC Protocol (8.7). Interpretation of this differential channel quality depends on the precoding feedback reporting mode. When the reporting mode is set to “SDMA,” the effect of SDMA is captured in the MIMO channel quality reported in r-mqch, which is a logical channel defined by the RCC MAC Protocol (8.7). In this case, the MIMO channel quality in the absence of SDMA is calculated by adding the differential

channel quality reported in the r-bfch to the channel quality reported in the r-mqich. When the reporting mode is set to “no SDMA,” the effect of SDMA is not captured in the MIMO channel quality reported in the r-mqich. In this case, the MIMO channel quality in the absence of SDMA is calculated by subtracting the differential channel quality value reported in the r-bfch from the MIMO channel quality reported in the r-mqich. In addition to the differential channel quality, the r-bfch report contains the index of the preferred precoding matrix selected by the access terminal.

Based on the preferred precoding matrix index and differential channel quality reports from different access terminals, the access network selects a set of access terminals to be superposed on the same channel resources. Power allocation and spectral efficiency of each access terminal is chosen according to the channel conditions.

The access network applies the selected precoding matrix to transmit to the chosen access terminals.

9.2.9 Rotational Orthogonal Frequency Division Multiplexing

Rotational Orthogonal Frequency Division Multiplexing is an optional scheme at the access terminal and at the access network. Rotational Orthogonal Frequency Division Multiplexing shall be used in the DRCH mode only.

When rotational Orthogonal Frequency Division Multiplexing is used, each set of D contiguous modulation symbols shall be rotated just before the inverse Fourier transform operation. The first set of D contiguous modulation symbols to be rotated shall use modulation symbols 0 through $D - 1$, the second set of modulation symbols shall use modulation symbols D through $2D - 1$, etc. The rotated symbols shall be generated according to Equation (25).

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_D \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & \dots & r_{D1} \\ r_{12} & r_{22} & \dots & r_{D2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1D} & r_{2D} & \dots & r_{DD} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} \quad (25)$$

where the $\mathbf{x}$ and $\mathbf{y}$ vectors represent the modulation symbols and the rotated symbols respectively. The matrix $\mathbf{R}_D$ represents the rotational code and D denotes the rotational dimension. The rotational dimension indicates the number of modulation symbols mapped to different subcarriers. For a rotational dimension of D , the rotational matrix $\mathbf{R}_D$ shall be given by Equation (26).

$$\mathbf{R}_D = \begin{bmatrix} \mathbf{R}_{D/2} \cos \theta_D & \mathbf{R}_{D/2} \sin \theta_D \\ -\mathbf{R}_{D/2} \sin \theta_D & \mathbf{R}_{D/2} \cos \theta_D \end{bmatrix} \quad (26)$$

The rotational matrices for rotational dimensions of 2 and 4 are given by Equation (27) and Equation (28).

$$\mathbf{R}_2 = \begin{bmatrix} r_{11} & r_{21} \\ r_{12} & r_{22} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (27)$$

$$\mathbf{R}_4 = \begin{bmatrix} r_{11} & r_{21} & r_{31} & r_{41} \\ r_{12} & r_{22} & r_{32} & r_{42} \\ r_{13} & r_{23} & r_{33} & r_{43} \\ r_{14} & r_{24} & r_{34} & r_{44} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \theta & \sin \theta \cos \theta & \cos \theta \sin \theta & \sin \theta \sin \theta \\ -\sin \theta \cos \theta & \cos \theta \cos \theta & -\sin \theta \sin \theta & \cos \theta \sin \theta \\ -\cos \theta \sin \theta & -\sin \theta \sin \theta & \cos \theta \cos \theta & \sin \theta \cos \theta \\ \sin \theta \sin \theta & -\cos \theta \sin \theta & -\sin \theta \cos \theta & \cos \theta \cos \theta \end{bmatrix} \quad (28)$$

The rotational angles and dimensions for Forward Data Channel are described in 9.4.1.2.5.1.

9.2.10 Subcarrier Allocation for Reverse Link Code Division Multiple Access subsegments and Reverse Orthogonal Frequency Division Multiple Access data channel

In this subclause, the procedures of the Reverse Link Code Division Multiple Access subsegments subcarrier allocation and hopping are described, as well as the procedures for Reverse Link OFDM data channel subcarrier allocation and hopping. Note that the subcarrier allocation changes every frame.

9.2.10.1 Hop-port definition and indexing

During the Reverse Link PHY frame portion of the transmission, the subcarriers of each Orthogonal Frequency Division Multiplexing symbol shall also use a second indexing scheme known as hop-port indexing. In this scheme, each Orthogonal Frequency Division Multiplexing symbol consists of $Q_{\text{SDMA,RL}} N_{\text{FFT}}$ individually-modulated hop-ports. Here $Q_{\text{SDMA,RL}}$ is equal to $\text{RLNumSDMADimensions}$, which is part of the public data of the Overhead Messages Protocol (see 11.6). The hop-ports are indexed from 0 through $Q_{\text{SDMA,RL}} N_{\text{FFT}} - 1$. There is a mapping between the $Q_{\text{SDMA,RL}} N_{\text{FFT}}$ hop-ports and the N_{FFT} subcarriers, called a hop-permutation. The hop-permutation may change as often as every Orthogonal Frequency Division Multiplexing symbol and is different for different sectors. The sequence of hop-permutations, also called the hopping sequence, is described in 9.2.10.²⁶

Note that the set of $Q_{\text{SDMA,RL}} N_{\text{FFT}}$ hop-ports shall be divided into $Q_{\text{SDMA,RL}}$ “SDMA subtrees,” indexed 0 through $Q_{\text{SDMA,RL}} - 1$. The notion of SDMA subtree is as defined in the RTC MAC Protocol (8.7). Each SDMA subtree shall have N_{FFT} hop-ports, and the SDMA subtree with index q shall contain hop-ports indexed $q N_{\text{FFT}}$ through $(q + 1) N_{\text{FFT}} - 1$.

9.2.10.2 Reverse Link hop pattern generation

Reverse Link hop pattern generation is a two step process as follows:

- a) Mapping hop-ports to subcarriers assuming nominal locations of Code Division Multiple Access subsegments.
- b) Relocating subcarriers that are displaced when Code Division Multiple Access subsegments hop from their nominal locations to actual locations.

9.2.10.3 Code Division Multiple Access Subsegments

The Reverse Link CDMA Control Channel and Reverse Link CDMA Data channels consist of one or more CDMA subsegments. Each CDMA subsegment consists of 128 contiguous subcarriers.

9.2.10.3.1 Code Division Multiple Access hopping zones

In this subclause, the set of Code Division Multiple Access hopping zones are defined. All Code Division Multiple Access subsegments are hopped among the Code Division Multiple Access hopping zones. The nominal location of each Code Division Multiple Access subsegment is defined. The subcarriers not belonging to guard or nominal locations of Code Division Multiple Access subsegments are defined as nominal available subcarriers.

A total of $N_{\text{CDMA-ZONES}} = \lceil (N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}}) / N_{\text{CDMA-SUBSEGMENT}} \rceil$ shall be defined. Let N_{BLOCK} denote the number of subcarriers in a tile. If $N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}} \leq 128$, then there is only one CDMA hopping zone with $f_{\text{START-CDMA}}(0) = 0$. Else, the Code Division Multiple Access hopping zones shall be indexed 0 through $N_{\text{CDMA-ZONES}} - 1$, and the Code Division Multiple Access zone with index k shall have subcarriers $f_{\text{START-CDMA}}(k)$ through $f_{\text{START-CDMA}}(k) + N_{\text{CDMA-SUBSEGMENT}} - 1$, where

²⁶The access terminal can compute the hop permutations for any given PHY frame and for any given sector using information available on the F-PBCCCH and F-SBCCCH of that sector.

$$f_{\text{START\_CDMA}}(k) = N_{\text{GUARD,LEFT,EFFECTIVE}} + N_{\text{BLOCK}} \left\lfloor \frac{k(N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}} - N_{\text{CDMA-SUBSEGMENT}})}{(N_{\text{CDMA-ZONES}} - 1)N_{\text{BLOCK}}} \right\rfloor$$

as illustrated in Figure 92. $N_{\text{CDMA-SUBSEGMENT}}$ is a constant of the Physical Layer Protocol.

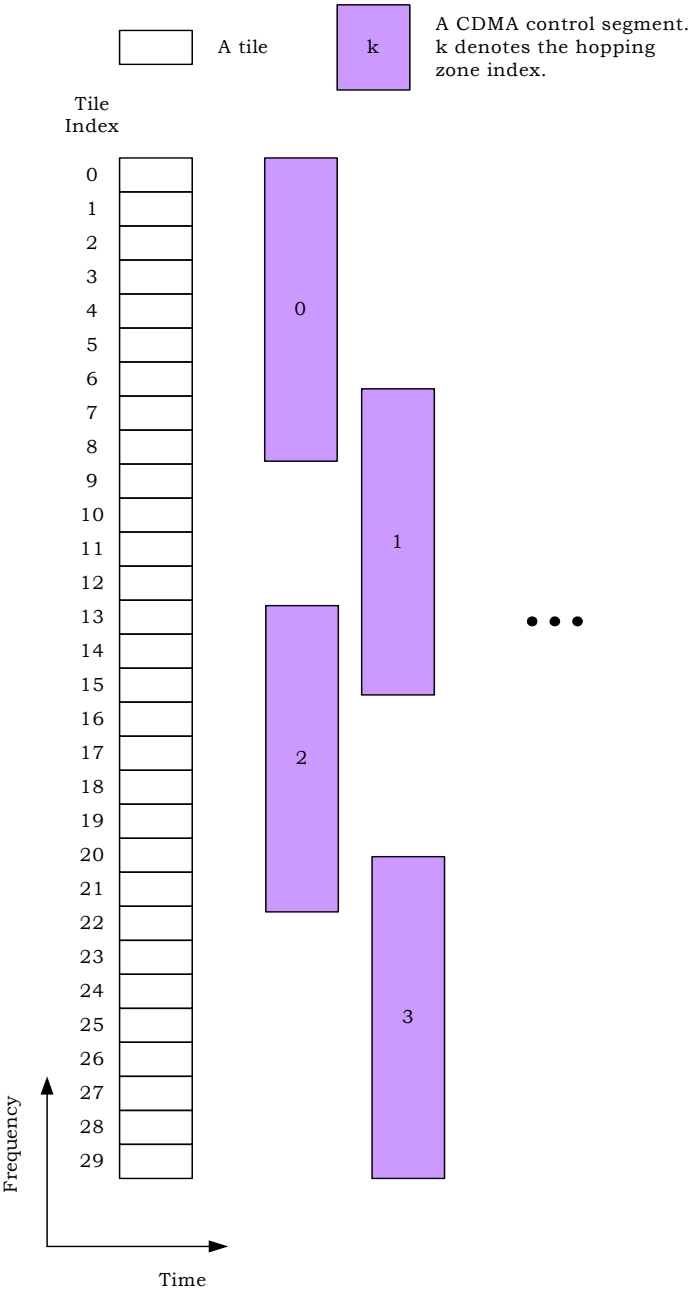


Figure 92—Illustration of Code Division Multiple Access hopping zones

9.2.10.3.2 Nominal Location of Code Division Multiple Access subsegments

The number of CDMA subsegments C in each Reverse Link PHY frame is determined by the $\text{CDMAInterlacesBitmap}$ and $\text{NumCDMASubsegments}_j$ fields of the Overhead Messages Protocol and are indexed 0 through $C - 1$, and j is the interlace index. The value of C may be interlace dependent.

The c^{th} Code Division Multiple Access subsegment shall nominally be mapped to the following:

- Code Division Multiple Access hopping zone c , when $(N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}})$ is an integer multiple of $N_{\text{CDMA-SUBSEGMENT}}$. In this case, the access network shall ensure that $C \leq N_{\text{CDMA-ZONES}}$.
- The CDMA hopping zone $2c$ when $(N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}})$ is not an integer multiple of $N_{\text{CDMA-SUBSEGMENT}}$. In this case the access network shall ensure that $\lfloor C \leq N_{\text{CDMA-ZONES}}/2 \rfloor$.

The subcarriers assigned to a CDMA hopping zone are described in 9.2.10.3.1. Note that this nominal mapping does not correspond to the actual set of subcarriers assigned to the CDMA subsegments. The nominal mapping is required to compute the data hopping pattern in subsequent subclauses.

9.2.10.3.3 Location of CDMA subsegments in subcarrier space

In the PHY frame with index j within the superframe i , the CDMA subsegment with index c shall be mapped to the CDMA hopping zone with index $k_{\text{TRUE}}(c)$, where

- $k_{\text{TRUE}}(c) = (\lfloor j_{\text{OVERALL}} / \text{NumRLInterlaces} \rfloor) + (j_{\text{OVERALL}} \bmod \text{NumRLInterlaces}) + c \bmod N_{\text{CDMA-ZONES}}$, when $(N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}})$ is an integer multiple of $N_{\text{CDMA-SUBSEGMENT}}$.
- $k_{\text{TRUE}}(c) = (\lfloor j_{\text{OVERALL}} / \text{NumRLInterlaces} \rfloor) + (j_{\text{OVERALL}} \bmod \text{NumRLInterlaces}) + c \times 2 \bmod N_{\text{CDMA-ZONES}}$ when $(N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}})$ is not an integer multiple of $N_{\text{CDMA-SUBSEGMENT}}$.

$j_{\text{OVERALL}} = (i \bmod 2^{33}) \times N_{\text{PHYFrames}} + j$ is the overall index of the PHY frame (i.e., not just the index within the superframe).

9.2.10.3.4 Nominally available subcarriers

All subcarriers that are not part of a nominal Code Division Multiple Access hopping zone (as defined above) and are not guard subcarriers shall be defined as “Nominally Available Subcarriers.” The nominally available subcarriers shall be sequentially indexed 0 through $N_{\text{AVAILABLE}} - 1$, where $N_{\text{AVAILABLE}} = \max(0, N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}} - C \times N_{\text{CDMA-SUBSEGMENT}})$.

9.2.10.3.5 Displaced subcarriers and newly-freed subcarriers

The set of subcarriers that belong to the actual locations of CDMA subsegments [CDMA hopping zones with $k_{\text{TRUE}}(c)$ as defined in 9.2.10.3.3] but not to the nominal location of any CDMA subsegment are known as “displaced subcarriers.” The displaced subcarriers are indexed sequentially from 0 through $N_{\text{DISPLACED}} - 1$, with lower indexed subcarriers also having a lower indexed “displaced subcarrier index.”

The set of subcarriers that belong to the nominal location of CDMA subsegments but not to the actual location of any CDMA subsegment shall be known as “newly freed subcarriers.” The newly freed subcarriers are indexed 0 through $N_{\text{NEWLYFREEDUP}} - 1$. Note that $N_{\text{DISPLACED}} = N_{\text{NEWLYFREEDUP}}$.

9.2.10.4 Subzones and usable hop-ports

9.2.10.4.1 Partition of hop-ports into subzones

In this subclause, the partitioning of hop-ports into subzones are described. The set of N_{FFT} hop-ports in each SDMA subtree shall be divided into $N_{\text{FFT}}/N_{\text{SUBZONE\_MAX,RL}}$ groups, referred to as “subzones.” The number of hop-ports in each subzone shall be $N_{\text{SUBZONE\_MAX,RL}}$, where $N_{\text{SUBZONE\_MAX,RL}}$ is the RLSubzoneSize field of the Overhead Messages Protocol and can be either 64 or 128. The hop-ports shall be allocated sequentially to the subzones i.e., hop-ports 0 through $N_{\text{SUBZONE\_MAX,RL}} - 1$ shall belong to subzone with index 0, hop-ports indexed $N_{\text{SUBZONE\_MAX,RL}}$ through $2N_{\text{SUBZONE\_MAX,RL}} - 1$ shall belong to subzone with index 1, etc.

Some or all hop-ports in a subzone may be usable (see 9.2.10.4.2 for the definition of usable hop-port). A subzone consisting of at least one usable hop-port is defined to be a usable subzone.

The number of usable subzones S shall be given by

$$S = \left\lceil \frac{N_{\text{AVAILABLE}}}{N_{\text{SUBZONE\_MAX,RL}}} \right\rceil$$

The subzones shall be indexed 0 through $S - 1$.

Let $S_{\text{SPLIT}} = (N_{\text{AVAILABLE}} / N_{\text{BLOCK}}) \bmod S$. Subzones indexed 0 through $(S_{\text{SPLIT}} - 1)$ shall have

$$N_{\text{SUBZONE-BIG}} = N_{\text{BLOCK}} \left\lceil \frac{N_{\text{AVAILABLE}}}{N_{\text{BLOCK}} S} \right\rceil \text{ usable hop-ports and subzones indexed } S_{\text{SPLIT}} \text{ through } S - 1$$

shall have $N_{\text{SUBZONE-SMALL}} = N_{\text{BLOCK}} \left\lfloor \frac{N_{\text{AVAILABLE}}}{N_{\text{BLOCK}} S} \right\rfloor$ usable hop-ports. The number of usable hop-ports

in subzone s shall be denoted as $N_{\text{SUBZONE}}(s)$.

9.2.10.4.2 Usable and unusable hop-ports

In this subclause, the notion of usable and unusable hop-ports are defined. This is done to ensure that the number of usable hop-ports (in a subtree) is equal to the available subcarriers. Thus, the total number of usable hop-ports in a subtree is equal to $(N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}} - C \times N_{\text{CDMA-SUBSEGMENT}})$.

For hop-port p , define $s = \left\lfloor \frac{p \bmod N_{\text{FFT}}}{N_{\text{SUBZONE\_MAX,RL}}} \right\rfloor$, and $r_p = (p \bmod N_{\text{SUBZONE\_MAX,RL}})$.

The hop-port p shall be usable only if all of the following conditions are true:

- $s < S$
- $r_p < N_{\text{SUBZONE}}(s)$

All other hop-ports shall be unusable.

A subzone that is comprised only of unusable hop-ports shall be referred to as an unusable subzone. All other subzones shall be usable.

9.2.10.4.3 Reverse Link Resource Channel structures

On the Reverse Link, a group of N_{BLOCK} hop-ports gets mapped to a contiguous group of N_{BLOCK} subcarriers. This mapping remains fixed for the duration of a Reverse Link PHY frame. The group of hop-ports shall be referred to as a “hop-port block” and the group of N_{BLOCK} subcarriers shall be referred to as a “subcarrier block.” The group of $N_{\text{BLOCK}} = 16$ hop-ports for the duration of $N_{\text{FRAME}} = 8$ Orthogonal Frequency Division Multiplexing symbols is also referred to as a “tile.”

The Reverse OFDMA Data Channel supports Global Hopping (GH) and Local Hopping (LH) structures. A given Reverse Link PHY frame uses either the GH structure or the LH structure. The primary difference between GH and LH structures is that in the LH structure a hop-port is constrained to hop within a “subzone,” while in the GH structure, a hop-port may hop over the entire bandwidth.

The set of Reverse Link PHY frames that use the GH structure are determined according to the following procedure:

- Define $N_{\text{DRCH}} = N_{\text{SUBZONE\_MAX\_FL}} \times \text{NumDRCHSubzones}$, where NumDRCHSubzones is a field of the Overhead Messages Protocol and $N_{\text{SUBZONE\_MAX\_FL}}$ is given by the FLSubzoneSize field of the Overhead Messages Protocol. N_{DRCH} represents the nominal number of subcarriers assigned to the DRCH zone on the Forward Link, as described in 9.2.14.4. Define the number of “interlaces” M_{GH} using the GH structure as $\text{round}(N_{\text{DRCH}} / (N_{\text{FFT}} - N_{\text{GUARD}}), 1)$. Each interlace is represented by an integer in the range $[0, 7]$.
- Define the set of interlaces I_{GH} using the GH structure as the empty set if $M_{\text{GH}} = 0$ and as the set of integers $\lfloor i \times \text{NumRLInterlaces} / M_{\text{GH}} \rfloor$, $0 \leq i < M_{\text{GH}}$ otherwise.
- A Reverse Link PHY frame with index j in superframe i shall use the GH structure if $(i \times N_{\text{PHYFrames}} + j) \bmod \text{NumRLInterlaces}$ belongs to the set I_{GH} , and shall use the LH structure otherwise.

9.2.10.5 Hop sequence generation for GH hop-ports

An alternative indexing scheme is used for each hop-port, which is based on its relative position in the SDMA subtree, the subzone in the subtree, the block within the subzone, and the hop-port within the block. A usable hop-port in this structure is mapped to a nominally available subcarrier according to a mapping function. The mapping function consists of a global permutation function $H_{\text{GLOBAL\_GH}}^{ij}$ common to all sectors and a sector-, SDMA subtree-, and subzone-specific permutation function $H_{\text{SECTOR\_GH}}^{ijqs}$. Both $H_{\text{GLOBAL\_GH}}^{ij}$ and $H_{\text{SECTOR\_GH}}^{ijqs}$ change every Reverse Link PHY frame. These permutations also repeat every 16 superframes i.e., the permutation in frame j of superframe i is the same as the permutation in frame j of superframe $(i + 16)$. A GH hop-port block is first permuted locally within the subzone it is in using $H_{\text{SECTOR\_GH}}^{ijqs}$. It is then further permuted globally by $H_{\text{GLOBAL\_GH}}^{ij}$ among the entire usable set of hop-port blocks in GH to arrive at a new block index. Since $H_{\text{GLOBAL\_GH}}^{ij}$ is the same across all sectors, the physical subcarriers allocated to a subzone are the same across all sectors. This is done in order to provide support for Fractional Frequency Reuse schemes. $H_{\text{SECTOR\_GH}}^{ijqs}$ is different across different sectors. This is done in order to provide interference diversity within a subzone.

9.2.10.5.1 Alternate indexing scheme for GH hop-ports

For convenience of notation, an alternative indexing scheme may be used for hop-ports in the GH structure. A hop-port p , in a Reverse Link PHY frame using the GH structure, shall be denoted as “hop-port (GH, q , s , b , r).” Here $q = \lfloor p / N_{\text{FFT}} \rfloor$ is the index of the SDMA subtree containing hop-port p , s is the subzone index containing hop-port p , b is the index of the block (within subzone s) containing this hop-port, and r is the index of the hop-port within the block. In this standard, the two notations shall be used interchangeably and “hop-port (GH, q , s , b , r)” shall be used to refer to hop-port $p = qN_{\text{FFT}} + sN_{\text{SUBZONE\_MAX\_RL}} + bN_{\text{BLOCK}} + r$.

9.2.10.5.2 Hop-port to subcarrier mapping for the GH structure

For each hop-port (GH, q, s, b, r) in Reverse Link PHY frame indexed j in superframe with index i, define a nominally available subcarrier index $f_{\text{AVAIL-GH}}(\text{GH}, q, s, b, r)$ as

$$f_{\text{AVAIL-GH}} = N_{\text{BLOCK}} H_{\text{GLOBAL, GH}}^{ij} (b_{\text{MIN}}(s) + H_{\text{SECTOR, GH}}^{ijqs}(b)) + r;$$

where

$b_{\text{MIN}}(s) = \left(\frac{1}{N_{\text{BLOCK}}} \right) \left(\sum_{i < s} N_{\text{SUBZONE}}(i) \right)$ is the number of usable blocks before subzone s

$H_{\text{GLOBAL, GH}}^{ij}$ is a permutation of size $N_{\text{AVAILABLE}}/N_{\text{BLOCK}}$, which shall be generated as described in 9.2.10.5.3

$H_{\text{SECTOR, GH}}^{ijqs}$ is a permutation of size $N_{\text{SUBZONE}}(s)/N_{\text{BLOCK}}$, which shall be generated as described in 9.2.10.5.4

If $f_{\text{AVAIL-GH}}$ is not a displaced subcarrier as defined in 9.2.10.3.5, then the hop-port (GH, q, s, b, r) shall be mapped to the subcarrier with nominally available subcarrier index $f_{\text{AVAIL-GH}}$.

If $f_{\text{AVAIL-GH}}$ is a displaced subcarrier as defined in 9.2.10.3.5, with displaced subcarrier index k, then the hop-port (GH, q, s, b, r) shall instead be mapped to a newly freed-up subcarrier with index k. Newly freed-up subcarriers are defined in 9.2.10.3.5.

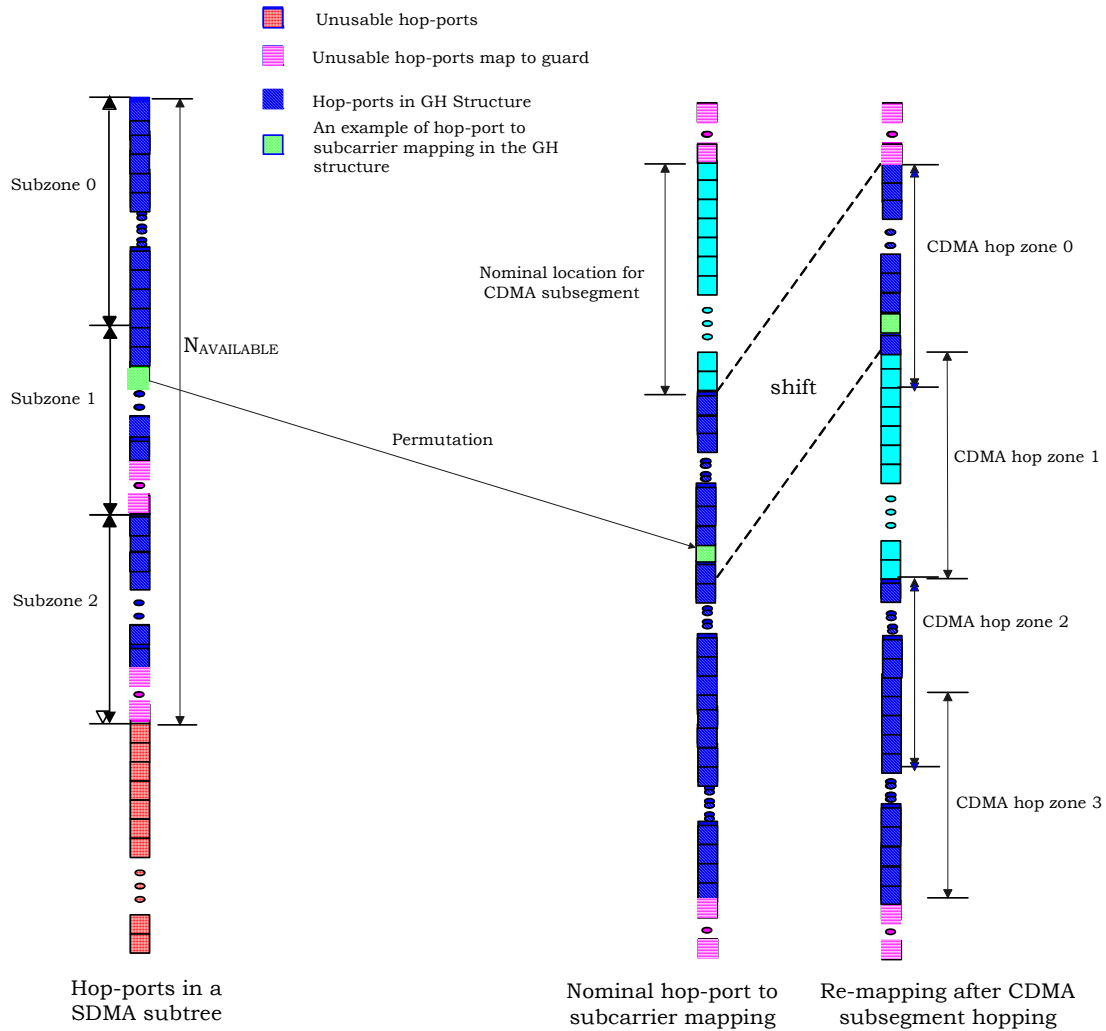


Figure 93—Illustration of hop-port to subcarrier mapping for the GH structure

9.2.10.5.3 Generation of $H_{\text{GLOBAL, GH}}^{ij}$

The permutation $H_{\text{GLOBAL, GH}}^{ij}$ is common to all sectors and shall be generated according to the following procedure.

- Set $\text{SEED}_{\text{GLOBAL}} = f_{\text{PHY-HASH}}(19 \times 32 \times 16 + (j \bmod 32) \times 16 + (i \bmod 16))$ where i is the superframe index, j is the frame index within the superframe and $f_{\text{PHY-HASH}}$ is the hash function described in 9.2.5.4.
- Let $H_{\text{GLOBAL, GH}}^{ij}$ is the permutation of size $N_{\text{AVAILABLE}}(s)/N_{\text{BLOCK}}$ generated using the common permutation generation algorithm described in 9.2.5.1 using seed $\text{SEED}_{\text{GLOBAL}}$.

9.2.10.5.4 Generation of $H_{\text{SECTOR, GH}}^{ijqs}$

The permutation $H_{\text{SECTOR, GH}}^{ijqs}$ varies from sector to sector and shall be generated according to the following procedure:

- Set $\text{SEED}_{\text{SECTOR, GH}} = f_{\text{PHY-HASH}}(1 \times 4 \times 2^{20} + \text{SectorSeed} \times 4 + (q \bmod 4))$ where SectorSeed is as defined in 9.2.2.3, and $f_{\text{PHY-HASH}}$ is the hash function described in 9.2.5.4.

- b) $H_{\text{SECTOR,GH}}^{ijqs}$ is the permutation of size $N_{\text{SUBZONE}}(s)/N_{\text{BLOCK}}$ generated using the common permutation generation algorithm described in 9.2.5.1 using seed SEED_{GH} , where s is the subzone index.

9.2.10.6 Hop sequence generation for Reverse Link PHY frames using the LH structure

A nominal mapping of a usable hop-port to an available subcarrier is first performed. A hop-port block is permuted locally within its subzone to obtain a mapped subcarrier block with the same subzone index. The permutation is a sector-, SDMA subtree-, and subzone-specific permutation function $H_{\text{SECTOR,LH}}^{ijqs}$. $H_{\text{SECTOR,LH}}^{ijqs}$ changes every frame and repeats every 16 superframes.

9.2.10.6.1 Alternate indexing scheme for LH structure

For convenience of notation, an alternative indexing scheme may be used for hop-ports in the LH structure. In this indexing scheme, a hop-port p shall be denoted as “hop-port (LH, q , s , b , r)” if the subzone s containing hop-port is allocated to the LH, where $q = \lfloor p/N_{\text{FFT}} \rfloor$ is the index of the SDMA subtree containing hop-port p , s is the index of the subzone containing hop-port p , b is the index of the block (within subzone s) containing this hop-port, and r is the index of the hop-port within the block. In this standard, the two notations shall be used interchangeably and “hop-port (LH, q , s , b , r)” shall be used to refer to hop-port $p = qN_{\text{FFT}} + sN_{\text{SUBZONE\_MAX,RL}} + bN_{\text{BLOCK}} + r$.

9.2.10.6.2 Hop-port to subcarrier mapping for the LH structure

For each hop-port (LH, q , s , b , r) in Reverse Link PHY frame indexed j within superframe indexed i , define an available subcarrier with index $f_{\text{AVAIL-LH}}$ as

$$f_{\text{AVAIL-LH}} = f_{\text{MIN-LH}}(s) + N_{\text{BLOCK}} H_{\text{SECTOR,LH}}^{ijqs}(b) + r$$

$$\text{where } f_{\text{MIN-LH}}(s) = \sum_{i < s} N_{\text{SUBZONE}}(i).$$

The permutation $H_{\text{SECTOR,LH}}^{ijqs}$ shall be generated according to the following procedure:

- a) Set $\text{SEED}_{\text{LH}} = f_{\text{PHY-HASH}}(18 \times 4 \times 2^{20} + \text{SectorSeed} \times 4 + (q \bmod 4))$ where SectorSeed is as defined in 9.2.2.2.3, $f_{\text{PHY-HASH}}$ is the hash function described in 9.2.5.4, and q is the SDMA subtree index.
- b) $H_{\text{SECTOR,LH}}^{ijqs}$ is the permutation of size $N_{\text{SUBZONE}}(s)/N_{\text{BLOCK}}$ generated with the common permutation generation algorithm described in 9.2.5.1 using seed SEED_{LH} , where s is the subzone index.

If the nominally available subcarrier with index $f_{\text{AVAIL-LH}}$ is not a displaced subcarrier as defined in 9.2.10.3.5, then the hop-port (LH, q , s , b , r) shall be mapped to the subcarrier with nominally available subcarrier index $f_{\text{AVAIL-LH}}$.

If the nominally available subcarrier with index $f_{\text{AVAIL-LH}}$ is a displaced subcarrier as defined in 9.2.10.3.5, with displaced subcarrier index k , then the hop-port (LH, q , s , b , r) shall instead be mapped to a newly freed-up subcarrier with index k . Newly freed-up subcarriers are defined in 9.2.10.3.5.

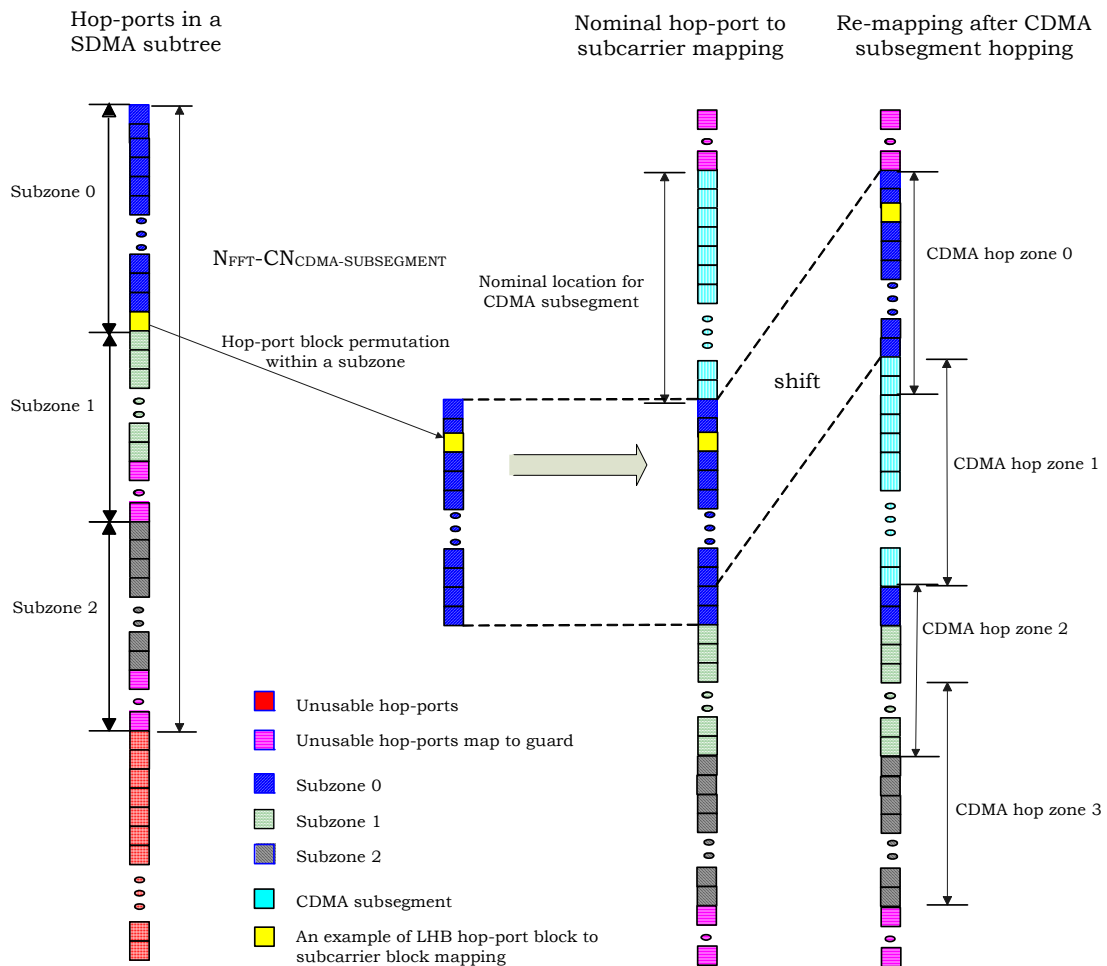


Figure 94—Illustration of hop-port to subcarrier mapping in the LH structure

9.2.11 Subcarrier allocation for Reverse Acknowledgment Channel

9.2.11.1 Reverse Acknowledgment Channel partial-tile definition

In this subclause, the procedure to choose the partial-tiles to carry the Reverse Acknowledgment Channel is described. All access terminals for which this sector is the Forward Link Serving Sector transmit their Reverse Acknowledgment Channel on these partial-tiles.

In each Reverse Link PHY frame, the Reverse Acknowledgment Channel shall be allocated multiple groups of $N_{\text{BLOCK,RACKCH}}$ contiguous subcarriers for the duration of $N_{\text{RACKCH-TILE-DURATION}}$ OFDM Symbols. The group of $N_{\text{BLOCK,RACKCH}}$ contiguous subcarriers shall be referred to as an Reverse Acknowledgment Channel block. The group of $N_{\text{BLOCK,RACKCH}}$ subcarriers spanning $N_{\text{RACKCH-TILE-DURATION}}$ OFDM symbols shall be referred to as an Reverse Acknowledgment Channel partial-tile. $N_{\text{BLOCK,RACKCH}}$ is a constant that takes the value 8, while $N_{\text{RACKCH-TILE-DURATION}}$ takes the value 8.

The number of Reverse Acknowledgment Channel partial-tiles allocated to the Reverse Acknowledgment Channel shall be $N_{\text{PARTIAL-TILES}}$, where $N_{\text{PARTIAL-TILES}} = 0$ if $\text{NumRACKNodes} = 0$, and is otherwise given by Equation (29).

$$N_{\text{PARTIAL-TILES}} = \max\left(\left\lceil \frac{\text{NumRACKNodes}}{8} \right\rceil, 4\right) \text{ if } N_{\text{FFT}} \geq 521, \quad (29)$$

where NumRACKNodes shall be specified by the RCC MAC Protocol (see 8.7). The set of subcarriers occupied by each Reverse Acknowledgment Channel partial-tile is specified in 9.2.11.2.

9.2.11.2 Reverse Acknowledgment Channel partial-tile selection

In the following algorithm, the permutation H_{RACKCH}^{ij} is used to randomly pick a set of tiles allocated to the Reverse Acknowledgment Channel. Within each tile, the term $(t_{\text{TILE}} \bmod 2) \times N_{\text{BLOCK,RACKCH}}$ selects either the upper or the lower portion.

The set of the Reverse Acknowledgment Channel partial-tiles of Reverse Link PHY frame j in superframe with index i shall be indexed from 0 to $N_{\text{PARTIAL-TILES}} - 1$ and shall be determined according to the following procedure:

- a) Set $\text{SEED}_{\text{RACKCH}} = f_{\text{PHY-HASH}}(23 \times 2^{20} + \text{SectorSeed})$. Here SectorSeed is as defined in 9.2.2.2.3 and $f_{\text{PHY-HASH}}$ is the common PHY hash function described in 9.2.5.4.
- b) Generate a permutation H_{RACKCH}^{ij} of size $(N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}})/N_{\text{BLOCK}}$ using the common permutation generation algorithm described in 9.2.5.1 with seed $\text{SEED}_{\text{RACKCH}}$, where $(N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}})/N_{\text{BLOCK}}$ is the number of usable subcarrier tiles in the entire bandwidth.
- c) For each t_{TILE} in the range $0 \leq t_{\text{TILE}} < N_{\text{PARTIAL-TILES}}$, define $f(t_{\text{TILE}})$ to be $H_{\text{RACKCH}}^{ij}(t_{\text{TILE}})$. Define $g(t_{\text{TILE}})$ to be equal to 0 if $f(t_{\text{TILE}}) \times N_{\text{BLOCK}} + N_{\text{GUARD,LEFT,EFFECTIVE}}$ is equal to $N_{\text{FFT}}/2$ and to be equal to $t_{\text{TILE}} \bmod 2$ otherwise. For each such t_{TILE} , the Reverse Acknowledgment Channel partial-tile with partial-tile index t_{TILE} shall constitute the set of subcarriers k in the range $f(t_{\text{TILE}}) \times N_{\text{BLOCK}} + g(t_{\text{TILE}}) \times N_{\text{BLOCK,RACKCH}} + N_{\text{GUARD,LEFT,EFFECTIVE}} \leq k < f(t_{\text{TILE}}) \times N_{\text{BLOCK}} + (g(t_{\text{TILE}})+1) \times N_{\text{BLOCK,RACKCH}} + N_{\text{GUARD,LEFT,EFFECTIVE}}$. Note that the first condition ensures that the Reverse Acknowledgment Channel partial-tile is not selected in such a way that it causes the Reverse Dedicated Pilot Channel on the remaining partial-tile to occupy the DC subcarrier. The time-duration of the Reverse Acknowledgment Channel partial-tile shall be given according to the following procedure:
 - Each Reverse Acknowledgment Channel partial-tile shall occupy the entire Reverse Link PHY frame, except for the first Reverse Link PHY frame in a superframe for which DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 0. In this case, the Reverse Acknowledgment Channel partial-tile shall occupy the OFDM symbols with indices 8–15 in this Reverse Link PHY frame.
- d) The Reverse Acknowledgment Channel allocation for this Reverse Link PHY frame shall consist all the partial-tiles defined above, namely those with indices $0 \leq t_{\text{TILE}} < N_{\text{PARTIAL-TILES}}$.

9.2.11.3 Reverse Acknowledgment Channel resource indexing

Each Reverse Acknowledgment Channel partial-tile shall be further divided into a number of Reverse Acknowledgment Channel subtiles, each of which spans $N_{\text{BLOCK,RACKCH}} = 8$ subcarriers in frequency and $N_{\text{RACKCH-SUBTILE-DURATION}} = 2$ Orthogonal Frequency Division Multiplexing symbols in time. The subtiles in each Reverse Acknowledgment Channel partial-tile shall be indexed from 0 to $N_{\text{SUBTILES}} - 1$, where $N_{\text{SUBTILES}} = N_{\text{RACKCH-TILE-DURATION}}/N_{\text{RACKCH-SUBTILE-DURATION}}$. The Orthogonal Frequency Division Multiplexing Symbol with index t within the partial-tile shall belong to subtile with index k_{SUBTILE} , where $k_{\text{SUBTILE}} = \lfloor t/N_{\text{RACKCH-SUBTILE-DURATION}} \rfloor$. Hence, each subtile allocated to the Reverse Acknowledgment Channel is thus indexed by the tuple $(t_{\text{TILE}}, k_{\text{SUBTILE}})$ where t_{TILE} is the Reverse Acknowledgment Channel partial-tile index and k_{SUBTILE} is the subtile index within that tile.

Each subtile has a total of $L = N_{\text{BLOCK,RACKCH}} N_{\text{RACKCH-SUBTILE-DURATION}} = 16$ subcarriers. Exponential sequences of length L , shall be used to modulate these subcarrier-symbols unless the all-zero sequence is

transmitted. The combination of a subtile and a specific exponential sequence of length L shall be referred to as a “Reverse Acknowledgment Channel resource.” A Reverse Acknowledgment Channel resource indexed by the tuple $(t_{\text{TILE}}, k_{\text{SUBTILE}}, \omega)$ shall correspond to the usage of exponential sequence E_{ω}^L to modulate the subtile $(t_{\text{TILE}}, k_{\text{SUBTILE}})$. Here the sequence E_{ω}^L is a sequence of length L , whose i^{th} element $E_{\omega}^L(i)$ is given by

$$E_{\omega}^L(i) = e^{-2\pi j \omega i / L}$$

where $0 \leq i < L$, and j denotes the complex number $(0,1)$.

9.2.12 Subcarrier allocation for Reverse OFDMA Dedicated Control Channel

9.2.12.1 Reverse OFDMA Dedicated Control Channel subcarrier allocation

If the Reverse OFDMA Dedicated Control Channel is present in a Reverse Link PHY frame, a number of tiles N_{TILES} may be allocated to the Reverse OFDMA Dedicated Control Channel. If $\text{NumRODCResources} = 0$, then the value of N_{TILES} shall be equal to 0 else the value of N_{TILES} shall be equal to $\max(2, \text{NumRODCResources}/2)$, where NumRODCResources is the number of the Reverse OFDMA Dedicated Control Channel reports in the Reverse Link PHY frame, and is determined for each Reverse Link PHY frame by the RCC MAC Protocol 8.7. The set of the Reverse OFDMA Dedicated Control Channel tiles of Reverse Link PHY frame j in superframe with index i shall be indexed from 0 to $N_{\text{TILES}} - 1$ and shall be determined according to the following procedure:

- a) Set $\text{SEED}_{\text{RODCCH}} = f_{\text{PHY-HASH}}(7 \times 2^{20} + \text{SectorSeed})$, where SectorSeed is defined in 9.2.2.2.3, and $f_{\text{PHY-HASH}}$ is the common PHY hash function described in 9.2.5.4.
- b) Generate a permutation H_{ij}^{RODCCH} of size $S_{\text{MAX}} = (N_{\text{FFT}} - N_{\text{GUARD,EFFECTIVE}}) / N_{\text{BLOCK}}$ using the common permutation generation algorithm described in 9.2.5.1 with seed $\text{SEED}_{\text{RODCCH}}$.
- c) Initialize counters k and t_{TILE} to 0.
- d) Repeat the following steps until $t_{\text{TILE}} = N_{\text{TILES}}$.
 - 1) Set $b = H_{ij}^{\text{RODCCH}}(k)$.
 - 2) Increment k by 1.
 - 3) Define $f = N_{\text{GUARD,LEFT,EFFECTIVE}} + b \times N_{\text{BLOCK}}$. If the set of subcarriers s in the range $f \leq s < f + N_{\text{BLOCK}}$ contains no subcarriers that have been allocated to the CDMA segment or to the Reverse Acknowledgement Channel in this Reverse Link PHY frame, then
 - i) Allocate the set of subcarriers f to $[f + N_{\text{BLOCK}} - 1]$ to the Reverse OFDMA Dedicated Control Channel for the duration of the Reverse Link PHY frame for all Reverse Link PHY frames other than those with index 0 when $\text{DuplexingMode} = \text{FDD}$ and $\text{FDDHalfDuplexTDDPartition} = 0$. For Reverse Link PHY frames with index 0 within the superframe when $\text{DuplexingMode} = \text{FDD}$ and $\text{FDDHalfDuplexTDDPartition} = 0$, allocate the set of subcarriers f to $[f + N_{\text{BLOCK}} - 1]$ only for the OFDM symbols indexed N_{PREAMBLE} through $N_{\text{PREAMBLE}} + N_{\text{FRAME}} - 1$ in the PHY frame.
 - ii) The Reverse OFDMA Dedicated Control Channel tile index of this tile shall be t_{TILE} .
 - iii) Increment t_{TILE} by 1.

9.2.12.2 Reverse OFDMA Dedicated Control Channel resource indexing

Each Reverse OFDMA Dedicated Control Channel tile shall further be divided into four Reverse OFDMA Dedicated Control Channel quarter-tiles, each of which spans $N_{\text{BLOCK}}/2$ subcarriers in frequency and $N_{\text{FRAME}}/2$ OFDM symbols in time. The quarter-tiles shall be indexed as $(t_{\text{TILE}}, k_{\text{QUARTER-TILE}})$. The quarter-tile with index $(t_{\text{TILE}}, k_{\text{QUARTER-TILE}})$ shall contain subcarriers f_k through $f_k + N_{\text{BLOCK}}/2 - 1$ in OFDM symbols t_k through $t_k + N_{\text{FRAME}}/2 - 1$ in the tile with Reverse OFDMA Dedicated Control Channel index t_{TILE} , where $f_k = (k_{\text{QUARTER-TILE}} \bmod 2) \times (N_{\text{BLOCK}}/2)$ and $t_k = \lfloor k_{\text{QUARTER-TILE}}/2 \rfloor (N_{\text{FRAME}}/2)$.

9.2.13 Reverse Link Silence Interval

The Reverse Link Silence Interval shall be defined as a group of subcarriers over a set of Reverse Link PHY frames. A Reverse Link PHY frame with index j in the superframe with index i belongs to the Reverse Link silence interval if $((i \times N_{\text{PHYFrames}}) + j) \bmod (N_{\text{PHYFrames}} \times \text{SilenceIntervalPeriod}) < \text{SilenceIntervalDuration}$, where, $\text{SilenceIntervalPeriod}$ and $\text{SilenceIntervalDuration}$ are parameters of the Overhead Messages Protocol, and $N_{\text{PHYFrames}}$ is the number of Reverse Link PHY frames in a Reverse Link superframe. If $\text{SilenceIntervalDuration} = 1$, and $i \bmod \text{SilenceIntervalPeriod} = 0$, and $j = 0$, and $\text{DuplexingMode} = \text{FDD}$ and $\text{FDDHalfDuplexTDDPartition} = 0$, then the first eight OFDM symbols with index 0 within superframe i shall be part of the Reverse Link Silence Interval. Otherwise, all eight OFDM symbols in that PHY frame j shall be part of the Reverse Link Silence Interval.

In an Orthogonal Frequency Division Multiplexing symbol that belongs to the Reverse Link Silence Interval, a subcarrier with index k shall belong to the Reverse Link Silence Interval if $\lfloor (k - N_{\text{GUARD,LEFT}}) / 128 \rfloor < \text{NumSilenceIntervalSubzones}$, where $\text{NumSilenceIntervalSubzones}$ is a parameter of the Overhead Messages Protocol.

The access terminal shall not transmit any energy on a subcarrier that is part of the Reverse Link Silence Interval. A variable called $N_{\text{GUARD,LEFT,EFFECTIVE}}$ shall be defined for each frame, where $N_{\text{GUARD,LEFT,EFFECTIVE}} = N_{\text{GUARD,LEFT}} + (128 \times \text{NumSilenceIntervalSubzones})$.

9.2.14 Forward Link Resource Channel structures and hop sequence generation

9.2.14.1 Hop-Port Indexing

For convenience of notation, the subcarriers of each Orthogonal Frequency Division Multiplexing symbol shall also use an alternative indexing scheme known as hop-port indexing. In this scheme, each Orthogonal Frequency Division Multiplexing symbol consists of a number of individually-modulated hop-ports. The number of hop-ports shall be equal to $Q_{\text{SDMA,FL}} N_{\text{HOP-PORTS-PER-SUBTREE}}$, where $N_{\text{HOP-PORTS-PER-SUBTREE}} = (N_{\text{DRCH}} + N_{\text{FFT}})$ in $\text{ResourceChannelMuxMode}$ 1 and equal to N_{FFT} in $\text{ResourceChannelMuxMode}$ 2, $N_{\text{DRCH-SUBZONES}}$ is equal to $N_{\text{SUBZONE, MAX\_FL}} \times N_{\text{DRCH-SUBZONES}}$, where $N_{\text{SUBZONE, MAX\_FL}}$ is equal to FLSubzoneSize , $N_{\text{DRCH-SUBZONES}}$ is equal to NumDRCHSubzones and $Q_{\text{SDMA,FL}}$ is equal to FLNumSDMASubtrees . FLSubzoneSize , NumDRCHSubzones and FLNumSDMASubtrees are parameters of the Overhead Messages Protocol.

The hop-ports are indexed from 0 through $Q_{\text{SDMA,FL}} N_{\text{HOP-PORTS-PER-SUBTREE}} - 1$. There is a mapping between the $Q_{\text{SDMA,FL}} N_{\text{HOP-PORTS-PER-SUBTREE}}$ hop-ports and the N_{FFT} subcarriers, called a hop-permutation. The hop-permutation may change every Orthogonal Frequency Division Multiplexing symbol and may be different for different sectors. The sequence of hop-permutations, also called the hop sequence, is described in 9.2.10.5 and 9.2.10.6.

Note that the set of $Q_{\text{SDMA,FL}} N_{\text{HOP-PORTS-PER-SUBTREE}}$ hop-ports is divided into $Q_{\text{SDMA,FL}}$ “SDMA subtrees,” indexed 0 through $Q_{\text{SDMA,FL}} - 1$, where the notion of SDMA subtree is as defined in the FTC MAC Protocol (see 8.6). Each SDMA subtree shall have $N_{\text{HOP-PORTS-PER-SUBTREE}}$ hop-ports, and the SDMA subtree with index q shall contain hop-ports indexed $q N_{\text{HOP-PORTS-PER-SUBTREE}}$ through $(q + 1) N_{\text{HOP-PORTS-PER-SUBTREE}} - 1$. Within each SDMA subtree, $N_{\text{HOP-PORTS-PER-SUBTREE}}$ hop-ports will be mapped to N_{FFT} subcarriers. In any OFDM symbol, hop-port to subcarrier mapping is the same for all $Q_{\text{SDMA,FL}}$ subtrees.

9.2.14.2 Forward Link Resource Channel structures

The Forward Data Channel supports Distributed Resource Channel (DRCH) and Block Resource Channel (BRCH) structures (see Figure 95).

9.2.14.2.1 Distributed Resource Channel (DRCH) structure

With the DRCH structure, a set of hop-ports is mapped to subcarriers that are scattered across the entire available bandwidth or across a large subset of the bandwidth. The set of subcarriers consists of N_{BLOCK} regularly spaced subcarriers on each of N_{FRAME} contiguous Orthogonal Frequency Division Multiplexing symbols.

9.2.14.2.2 Block Resource Channel (BRCH) structure

In the BRCH structure, users are assigned sets of N_{BLOCK} contiguous subcarriers. The mapping between hop-ports and frequency is kept constant throughout the PHY frame (N_{FRAME} OFDM symbols). Each set, therefore, defines a hop region consisting of N_{BLOCK} contiguous subcarriers and N_{FRAME} contiguous Orthogonal Frequency Division Multiplexing symbols, also referred to as a tile.

9.2.14.3 Multiplexing Resource Channel structures

DRCH and BRCH structures can both be used in the same PHY frame so that frequency diversity and frequency-selective transmissions are simultaneously supported. There are two multiplexing modes defined by the value of ResourceChannelMuxMode, which may be equal to 1 or 2. The choice of resource multiplexing mode is based on the parameter ResourceChannelMuxMode in the Overhead Messages Protocol. If ResourceChannelMuxMode = 1, the DRCH structures are punctured onto the BRCH structures. The common pilot is transmitted over the entire bandwidth. If ResourceChannelMuxMode = 2, the DRCH and BRCH structures are only used on different subzones where subzones are defined in 9.2.14.4.1. The size of a subzone is configurable by the access network (see Figure 96).

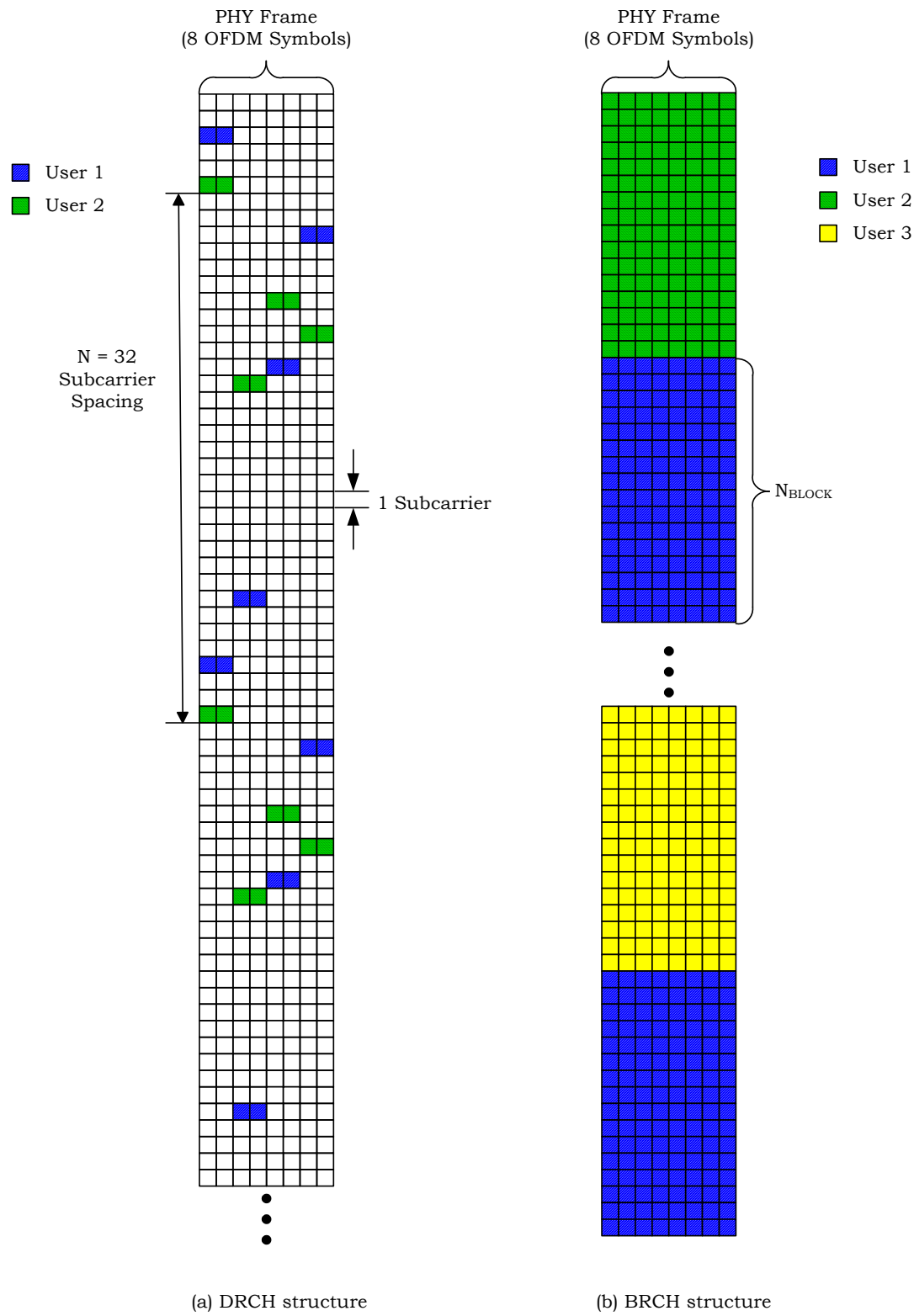
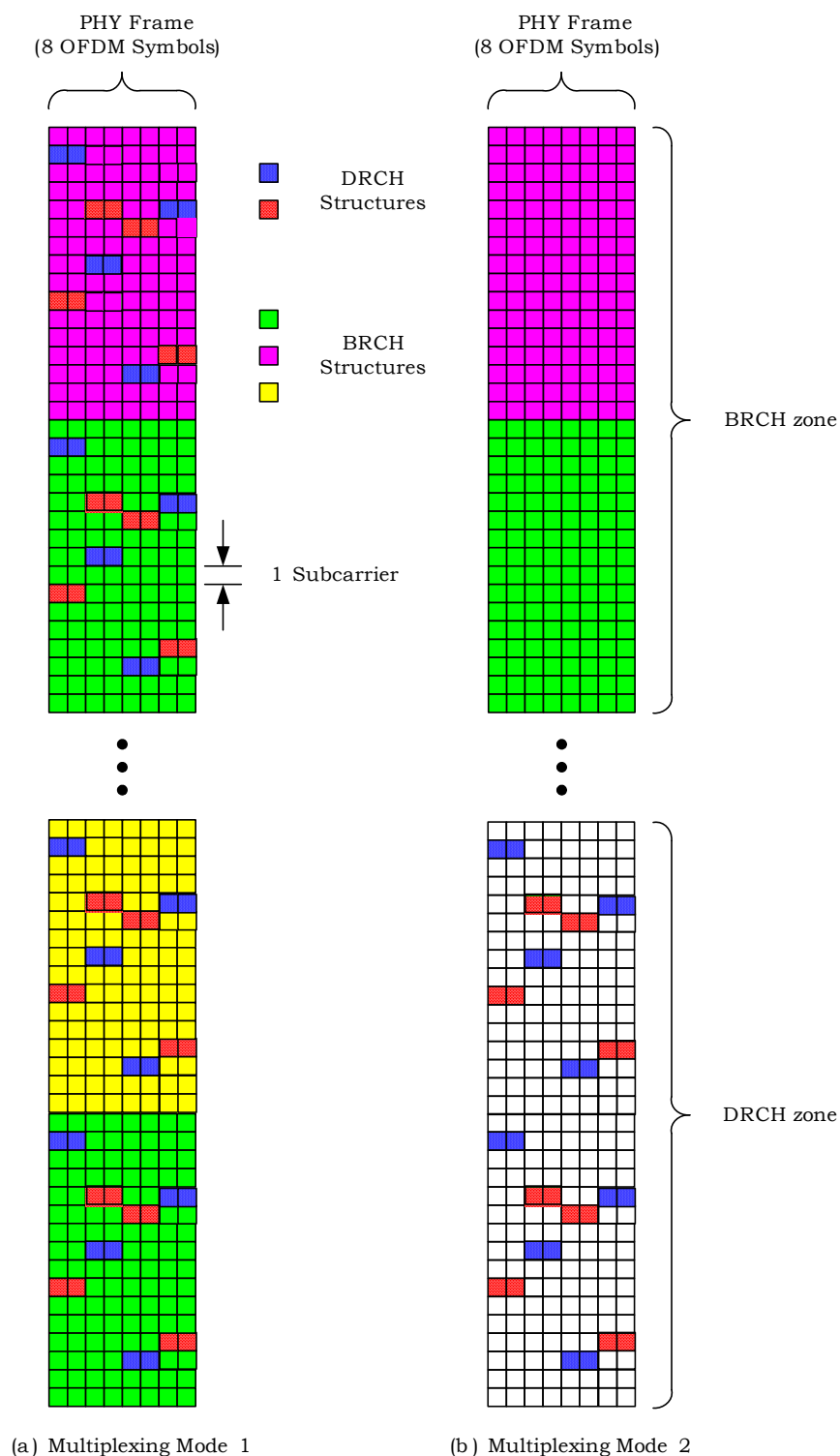


Figure 95—Examples of DRCH and BRCH structures

**Figure 96—Example of Multiplexing Resource structure**

9.2.14.4 Hop-port partitioning

9.2.14.4.1 Parameters for subzone partitioning

In this subclause, two parameters S and $N_{\text{SUBZONE}}(s)$ related to subzones are defined. Define S as

$$S = \left\lceil \frac{N_{\text{FFT}} - N_{\text{GUARD}}}{N_{\text{SUBZONE, MAX}}} \right\rceil$$

where $N_{\text{SUBZONE, MAX, FL}}$ is given by the FLSubzoneSize field of the Overhead Messages Protocol and can be either 64 or 128.

Let $S_{\text{SPLIT}} = ((N_{\text{FFT}} - N_{\text{GUARD}}) / N_{\text{BLOCK}}) \bmod S$. If $0 \leq s < S_{\text{SPLIT}}$ define

$$N_{\text{SUBZONE}}(S) = N_{\text{BLOCK}} \left\lceil \frac{N_{\text{FFT}} - N_{\text{GUARD}}}{N_{\text{BLOCK}} S} \right\rceil$$

If $S_{\text{SPLIT}} \leq s < S$ define

$$N_{\text{SUBZONE}}(S) = N_{\text{BLOCK}} \left\lceil \frac{N_{\text{FFT}} - N_{\text{GUARD}}}{N_{\text{BLOCK}} S} \right\rceil$$

9.2.14.4.2 Reserved subzones

A number of subzones shall be reserved in certain Forward Link PHY frames. A reserved subzone always maps to a contiguous set of subcarriers. A hop-port from any reserved subzone is referred to as reserved hop-port.

If $k = (i \times N_{\text{PHYFrames}} + j) \bmod \text{NumFLInterlaces}$ is a reserved interlace, then the number of reserved subzones in PHY frame j of superframe i shall equal $N_{\text{RESERVED-SUBZONES}}(k)$, where $N_{\text{RESERVED-SUBZONES}}(k)$ is equal to the NumFLReservedSubzones field of the Overhead Messages Protocol. The set of reserved subzones shall be $S - N_{\text{RESERVED-SUBZONES}}(k)$ through $S - 1$. The subcarriers to which these BRCH subzones map to shall be the reserved subcarriers. The mapping of hop-ports to subcarriers is as specified in 9.2.14.4.

The reserved subzones shall be indexed 0 through $N_{\text{RESERVED-SUBZONES}}(k)$ in increasing order (i.e., reserved subzone 0 is hop-port subzone $S - N_{\text{RESERVED-SUBZONES}}(k)$, reserved subzone 1 is hop-port subzone $S - N_{\text{RESERVED-SUBZONES}}(k) + 1$ etc.).

The mapping from reserved hop-ports in the reserved subzones shall be sequential i.e., if a subzone s is reserved and the 0th hop-port of the subzone s maps to the subcarrier f (as described in 9.2.14.5), then the n^{th} hop-port within subzone s maps to subcarrier $f + n$, where $0 \leq n < N_{\text{SUBZONE}}(s)$.

9.2.14.4.3 Alternate indexing of hop-ports

In this subclause, hop-ports are divided into subzones and an alternative indexing scheme is described. The hop-ports of each SDMA subtree are evenly divided into groups with size $N_{\text{SUBZONE, MAX, FL}}$, referred to as “hop-port subzones.” The hop-port subzones are further grouped into DRCH zone and BRCH zone. An alternative indexing scheme is used for each hop-port, which is based on its relative position in the SDMA subtree, the subzone in the subtree, the block within the subzone, and the hop-port within the block.

For convenience of notation, an alternative indexing scheme may be used for hop-ports. A hop-port p shall be denoted as “hop-port (DRCH, q , s , b , r)” if the hop-port is a DRCH hop-port and as “hop-port (BRCH, q , s , b , r)” if it is a BRCH hop-port. The values of q , s , b , and r shall be computed as follows:

- a) $\lfloor q = p / N_{\text{HOP-PORTS-PER-SUBTREE}} \rfloor$.
- b) Define $N_{\text{EFFECTIVE-DRCH-SUBZONES}}$ as
 - $\min(N_{\text{DRCH-SUBZONES}}, S - N_{\text{RESERVED-SUBZONES}}(k))$, where k is defined in 9.2.14.4.2 and $N_{\text{DRCH-SUBZONES}}$ is equal to the NumDRCHSubzones field of the Overhead Messages Protocol.
- c) Hop-ports 0 through $N_{\text{SUBZONE\_MAX\_FL}} \times N_{\text{EFFECTIVE-DRCH-SUBZONES}} - 1$ within the subtree shall be DRCH hop-ports. All other hop-ports shall be BRCH hop-ports.
- d) s shall be given by
 - 1) $\lfloor (p \bmod N_{\text{HOP-PORTS-PER-SUBTREE}}) / N_{\text{SUBZONE\_MAX\_FL}} \rfloor$ if ResourceChannelMuxMode = 2 and for DRCH hop-ports if ResourceChannelMuxMode = 1.
 - 2) $\lfloor ((p \bmod N_{\text{HOP-PORTS-PER-SUBTREE}}) - N_{\text{DRCH}}) / N_{\text{SUBZONE\_MAX\_FL}} \rfloor$ for BRCH resources if ResourceChannelMuxMode = 1.
- e) $b = \lfloor (p \bmod N_{\text{SUBZONE\_MAX\_FL}}) / N_{\text{BLOCK}} \rfloor$.
- f) $r = p \bmod N_{\text{BLOCK}}$.

Thus q is the index of the SDMA subtree, s is the index of the subzone within the subtree, b is the index of the hop-port-block within the subzone and r is the index of the hop-port within the block.

In this standard, the notations p and (DRCH/BRCH, q , s , b , r) are used interchangeably.

9.2.14.4.4 Usable and unusable hop-ports

Since the number of available subcarriers is less than N_{FFT} , the hop-ports are classified into usable and unusable hop-ports in this subclause. The goal is to ensure that the number of usable hop-ports is equal to the number of available subcarriers. A hop-port is marked as usable only if it is not a reserved hop-port and it does not map to a guard subcarrier.

A hop-port (DRCH/BRCH, q , s , b , r) shall be usable only if all of the following conditions are true:

- $s \leq S < S - N_{\text{RESERVED-SUBZONES}}(k)$.
- $b \times N_{\text{BLOCK}} + r < N_{\text{SUBZONE}}(s)$.

All other hop-ports shall be unusable. All unusable hop-ports shall be mapped to the guard subcarriers. The exact mapping is not specified since the unusable hop-ports shall not be modulated. Note that the number of usable hop-ports in subzone s is $N_{\text{SUBZONE}}(s)$.

9.2.14.5 Hop sequence generation for PHY frames

In each SDMA subtree, a usable hop-port is mapped to an available subcarrier that is not a guard or a reserved subcarrier. BRCH hop-port mapping is performed first, followed by DRCH hop-port mapping.

For BRCH hop-port mapping, a hop-port block of N_{BLOCK} contiguous hop-ports is mapped to a block of contiguous subcarriers of the same size. The mapping function consists of a sector- and subzone-specific permutation function $H^{\text{its}}_{\text{SECTOR, BRCH}}$. It permutes the hop-port blocks locally within a subzone. $H^{\text{its}}_{\text{SECTOR, BRCH}}$ changes every frame and repeats every 16 superframes i.e., the permutation in Orthogonal Frequency Division Multiplexing symbol t in superframe i is the same as the permutation in Orthogonal Frequency Division Multiplexing symbol t of superframe $(i + 16)$. Examples of BRCH hop-port to subcarrier mapping in for ResourceChannelMuxMode = 1 and ResourceChannelMuxMode = 2 are shown in Figure 97 and Figure 98, respectively.

The DRCH hop-port mapping is defined by a sector and subzone-specific offset $\text{InnerOffset}_{\text{DRCH}}$ and pruned bit reversal interleaver, which is described in 9.2.5.2. $\text{InnerOffset}_{\text{DRCH}}$ changes every two Orthogonal Frequency Division Multiplexing symbols and repeats every 16 superframes. It maps a hop-port block of N_{BLOCK} contiguous hop-ports to N_{BLOCK} subcarriers regularly spaced over the entire DRCH available subcarriers. The starting subcarriers of the DRCH blocks within the DRCH zone are chosen in pruned bit reversed order. The sector and subzone-specific offset is then applied to the starting subcarriers of the DRCH blocks within a subzone to randomize interference from other sectors in that DRCH subzone. The starting subcarrier of the entire DRCH zone changes every two OFDM symbols in order to sample the whole set of available subcarriers. If $\text{ResourceChannelMuxMode} = 1$, the DRCH available subcarriers are all subcarriers except for the guard subcarriers or the subcarriers mapped by a reserved subzone. If $\text{ResourceChannelMuxMode} = 2$, the DRCH available subcarriers are subcarriers within a DRCH zone, separated from the BRCH zone. If $\text{ResourceChannelMuxMode} = 1$, a subcarrier mapped by both a BRCH hop-port and a DRCH hop-port shall be mapped to the DRCH hop-port. In this case, the BRCH subcarrier block is punctured by this DRCH subcarrier. Figure 97 and Figure 98 show an example of DRCH hop-port to subcarrier mapping for $\text{ResourceChannelMuxMode} = 1$ and for $\text{ResourceChannelMuxMode} = 2$, respectively.

Finally, the mapping function is the same for all SDMA subtrees, i.e., the i^{th} hop-port within each SDMA subtree is mapped to the same subcarrier.

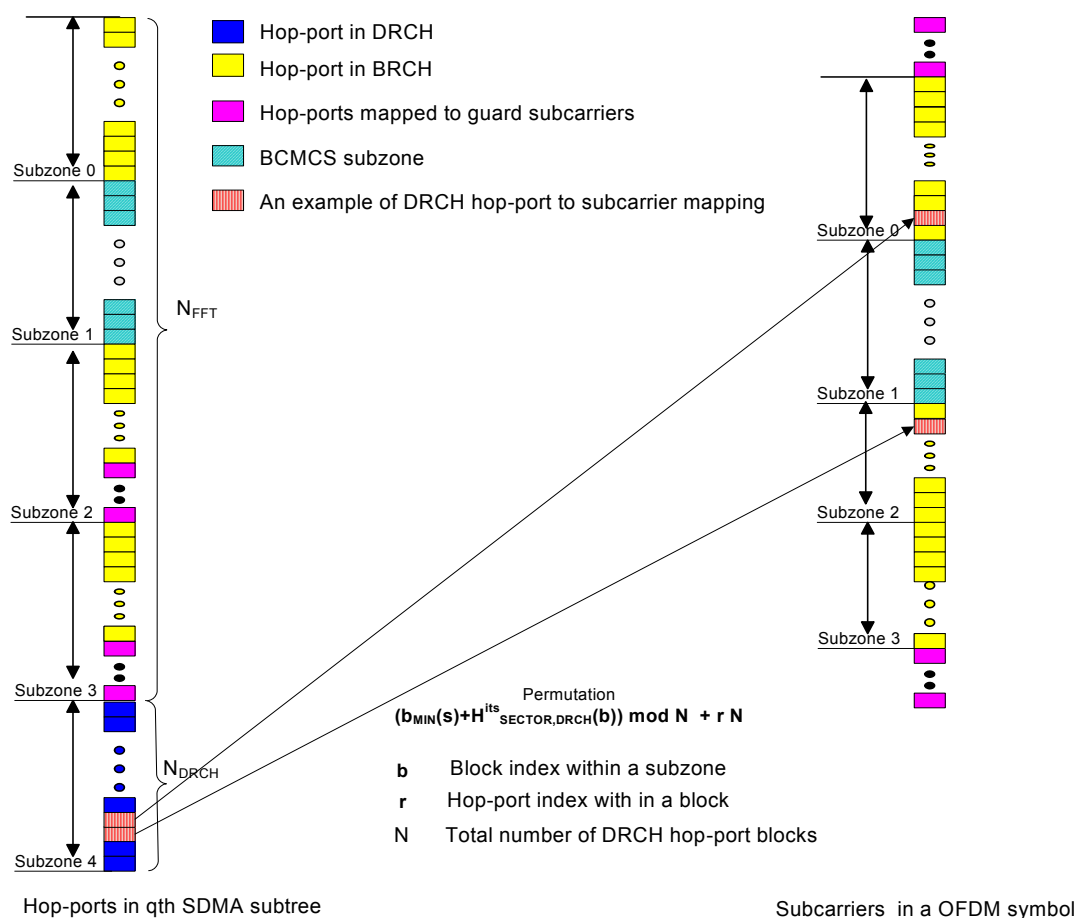


**Figure 97—Illustration of DRCH Hop-port to subcarrier mapping if
ResourceChannelMuxMode = 1**

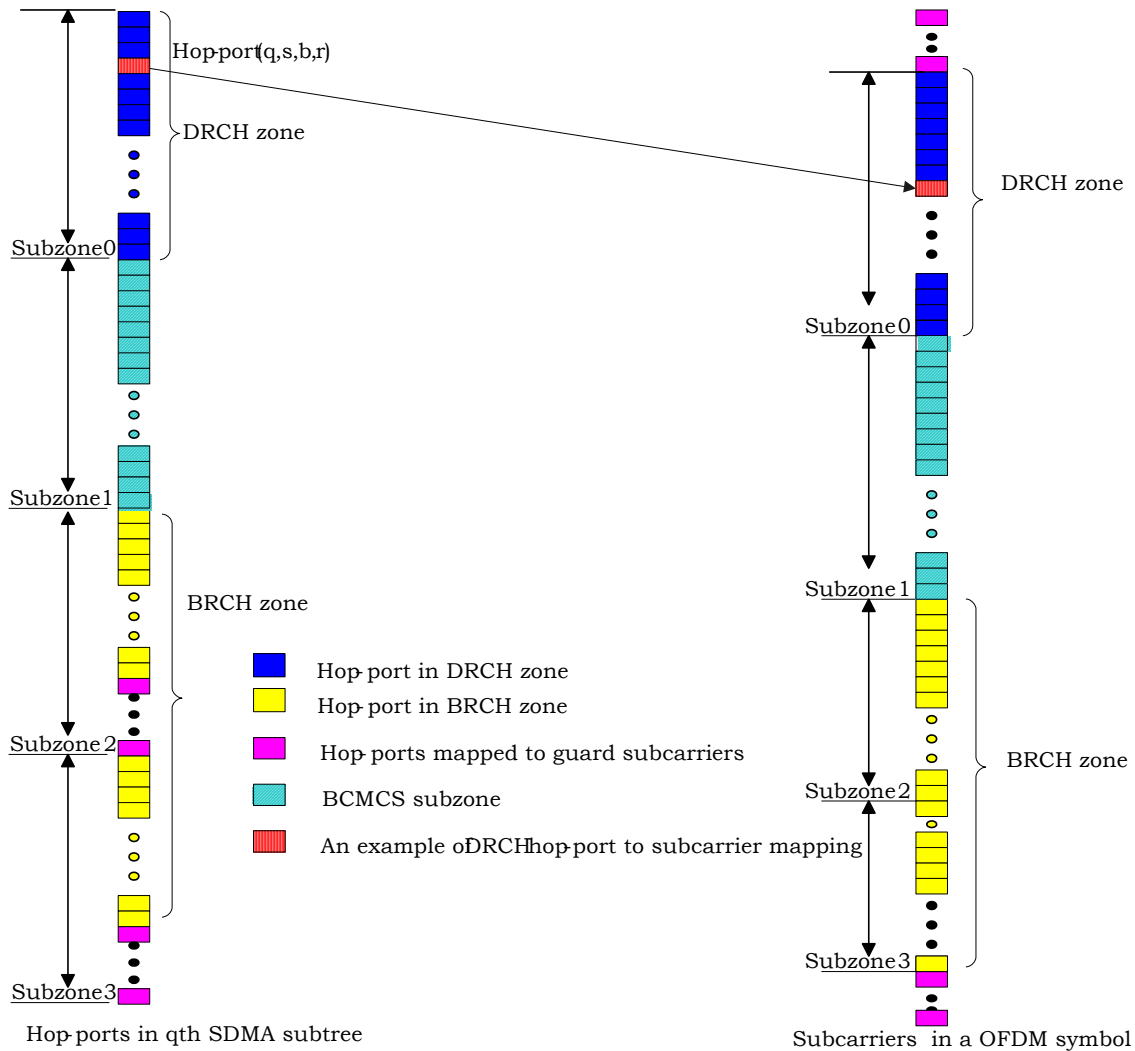


**Figure 98—Illustration of DRCH hop-port to subcarrier mapping in
ResourceChannelMuxMode = 2**

9.2.14.5.1 BRCH hop-port to subcarrier mapping

If ResourceChannelMuxMode = 1, a usable hop-port (BRCH, q, s, b, r) in Forward Link PHY frame indexed j within superframe indexed i , shall be mapped to subcarrier indexed

$$f = N_{\text{GUARD,LEFT}} + \sum_{\{k < s\}} N_{\text{SUBZONE}}(k) + H_{\text{SECTOR}}^{ijs}(b) \times N_{\text{BLOCK}} + r$$

where H_{SECTOR}^{ijs} is intra-subzone permutation obtained as follows:

- Set $\text{SEED}_{\text{SECTOR}} = f_{\text{PHY-HASH}}(6 \times 2^{20} + \text{SectorSeed})$ where SectorSeed is as defined in 9.2.2.2.3 and $f_{\text{PHY-HASH}}$ is the hash function described in 9.2.5.4.
- H_{SECTOR}^{ijs} is the permutation of size $N_{\text{SUBZONE}}(s)/N_{\text{BLOCK}}$ generated with seed $\text{SEED}_{\text{SECTOR}}$ using the common permutation generation algorithm described in 9.2.5.1.

If a reserved hop-port maps to subcarrier f , then BRCH hop-port (BRCH, q , s , b , r) is not a usable BRCH hop-port, unless it is used for the transmission of the Forward Superposed Data Channel.

If ResourceChannelMuxMode = 2 (see Figure 99), a BRCH hop-port (BRCH, q , s , b , r) in Forward Link PHY frame indexed j within superframe indexed i shall be mapped to subcarrier indexed f , which is defined according to the following procedure:

- If $N_{\text{EFFECTIVE-DRCH-SUBZONES}} = 0$ and $N_{\text{RESERVED-SUBZONES}}(k) = 0$, define $\pi[u] = u$, $0 < u < S$.
- If $N_{\text{EFFECTIVE-DRCH-SUBZONES}} > 0$ or if $N_{\text{RESERVED-SUBZONES}}(k) > 0$, define $\pi(u) = (\text{PBRI}(u, S) + ((N_{\text{FLPHYFrames}} \times i + j) \bmod \text{NumFLInterlaces}) \bmod S)$ if the BRCHSubzoneCyclingEnabled parameter of the Overhead Messages Protocol is set to 1 and $\pi(u) = \text{PBRI}(u, S)$ otherwise, where the function PBRI is defined in 9.2.5.2.
- Compute $f = N_{\text{GUARD,LEFT}} + \sum_{\{\pi(u) < (s)\}} N_{\text{SUBZONE}}(u) + H_{\text{SECTOR}}^{\text{ijs}}(b) \times N_{\text{BLOCK}} + r$

where $H_{\text{SECTOR}}^{\text{ijs}}$ is intra-subzone permutation obtained as follows:

- Set $\text{SEED}_{\text{SECTOR}} = f_{\text{PHY-HASH}}(6 \times 2^{20} + \text{SectorSeed})$ where SectorSeed is as defined in 9.2.2.2.3 and $f_{\text{PHY-HASH}}$ is the hash function described in 9.2.5.4.
- $H_{\text{SECTOR}}^{\text{ijs}}$ is the permutation of size $N_{\text{SUBZONE}}(s)/N_{\text{BLOCK}}$ generated with seed $\text{SEED}_{\text{SECTOR}}$ using the common permutation generation algorithm described in 9.2.5.1.

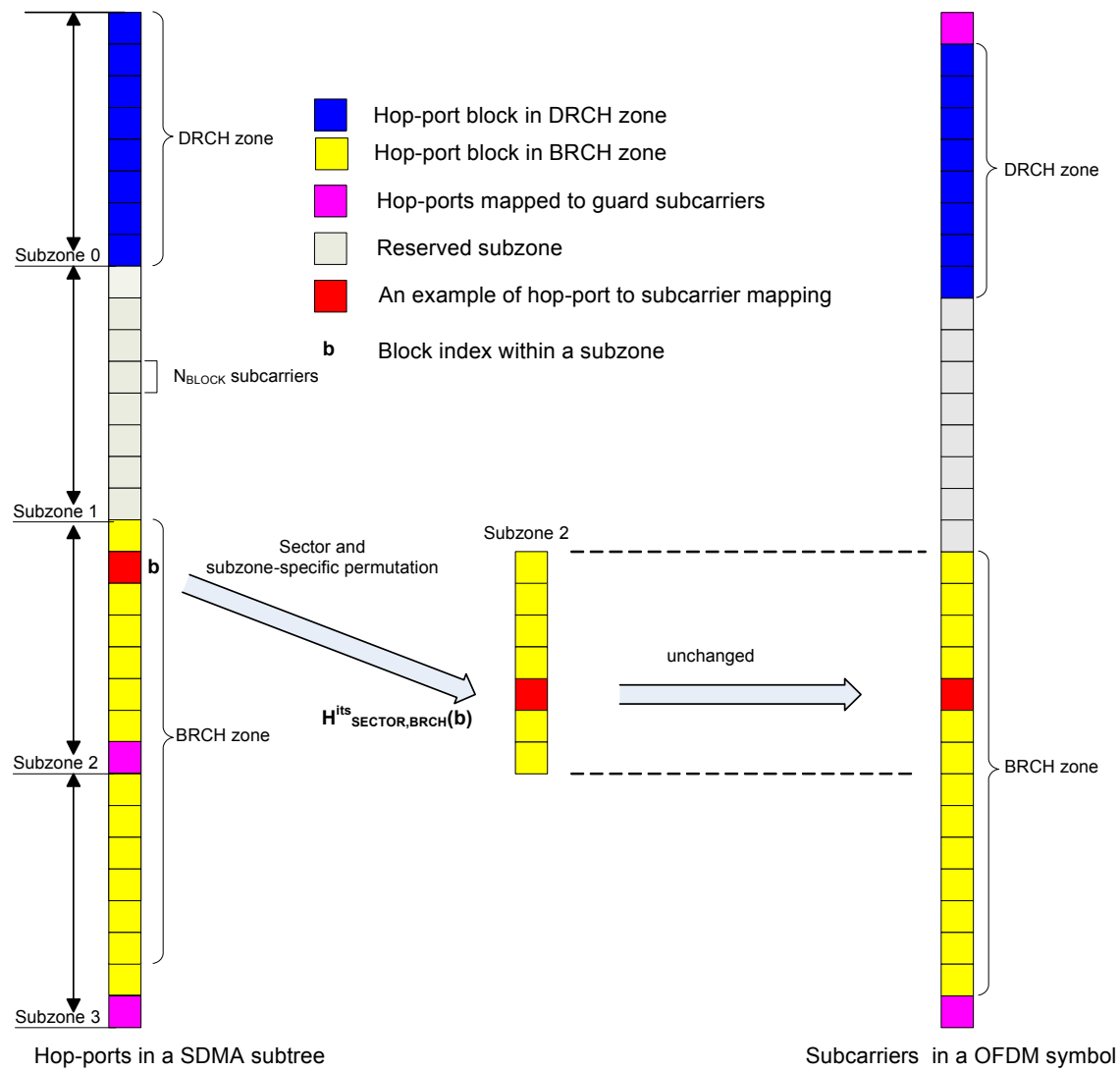


Figure 99—Illustration of BRCH hop-port to subcarrier mapping if ResourceChannelMuxMode = 2

9.2.14.5.2 DRCH hop-port to subcarrier mapping

9.2.14.5.2.1 DRCH Available Subcarrier Indexing

For convenience of notation, the subcarriers are indexed by a “DRCH available subcarrier index.” The available subcarrier index shall be defined as follows:

- a) Initialize a subcarrier counter f and an available subcarrier index counter $f_{AVAIL-DRCH}$ to 0.
- b) Repeat the following steps until $f = N_{FFT}$:
 - 1) Set a flag $FLAG_{AVAIL}$ to TRUE.
 - 2) Set $FLAG_{AVAIL}$ to FALSE if the subcarrier with index f is
 - i) A guard subcarrier, or
 - ii) A subcarrier mapped by a reserved hop-port.

- iii) If the ResourceChannelMuxMode is 2 and a BRCH hop-port is mapped to this subcarrier via the mapping described in 9.2.10.6.
- 3) If FLAG_{AVAIL} is true,
 - i) The DRCH-available-subcarrier-index of subcarrier f is defined as INDEX_{AVAIL-DRCH}.
 - ii) Increment INDEX_{AVAIL-DRCH} by 1.
- 4) Increment f by 1.
- c) The total number of DRCH available subcarriers N_{DRCH-AVAIL-SUBCARRIERS} shall be equal to the final value of the counter INDEX_{AVAIL-DRCH}.

Note that the DRCH available subcarrier index of a given subcarrier may change every frame. Also note that the subcarrier indexing is different in the two ResourceChannelMuxModes.

9.2.14.5.2.2 DRCH hop-port to subcarrier mapping

Compute the minimum spacing between DRCH subcarriers N_{MIN-DRCH-SPACING} and the number of occurrences N_{MAX-DRCH-SPACING} of spacing (N_{MIN-DRCH-SPACING} + 1) as follows: N_{MIN-DRCH-SPACING} = $\lfloor N_{\text{DRCH-AVAIL-SUBCARRIERS}} / N_{\text{DRCH-USABLE}} \rfloor$, N_{MAX-DRCH-SPACINGS} = (N_{DRCH-AVAIL-SUBCARRIERS} mod N_{DRCH-USABLE}) / N_{BLOCK}, where

$$N_{\text{DRCH-USABLE}} = \sum_{i < N_{\text{EFFECTIVE-DRCH-SUBZONES}}} N_{\text{SUBZONE}}(i) \text{ is the total number of usable DRCH hop-ports per subtree.}$$

In the Orthogonal Frequency Division Multiplexing symbol with index t in superframe with index i, the hop-port (DRCH, q, s, b, r) shall be mapped to the subcarrier with DRCH available subcarrier index INDEX_{AVAIL-DRCH}(DRCH, q, s, b, r) defined according to the following procedure:

- a) Compute a common offset ZoneOffset_{DRCH} that is applied to the entire DRCH zone and changes every two Orthogonal Frequency Division Multiplexing symbols: ZoneOffset_{DRCH} = f_{PHY=HASH}(18 × 16 × 128 + (i mod 16) × 128 + $\lfloor t/2 \rfloor$) mod (N_{DRCH-AVAIL-SUBCARRIERS} / N_{BLOCK}) if ResourceChannelMuxMode = 1 and ZoneOffset_{DRCH} = 0 otherwise.
- b) Compute the number of usable subcarrier blocks comprised of all subzones preceding subzone s:

$$\text{SubzoneOffset}_{\text{DRCH}} = \left(\frac{1}{N_{\text{BLOCK}}} \right) \sum_{i < s, i \in \text{DRCH}} N_{\text{SUBZONE}}(i)$$

- c) Compute a subzone specific and sector specific offset InnerOffset_{DRCH} within the subzone that changes every two Orthogonal Frequency Division Multiplexing symbols: InnerOffset_{DRCH} = f_{PHY=HASH}(17 × 2²⁰ × 128 + SectorSeed × 128 + $\lfloor t/2 \rfloor$) mod (N_{SUBZONE}(s) / N_{BLOCK}).
- d) Compute reference position RefPos_{DRCH} of the usable subcarrier block b as a pruned bit reversed value of its logical location corresponding to the subzone s and block index b:

$$\text{RefPos}_{\text{DRCH}} = \text{PBRI}(\text{SubzoneOffset}_{\text{DRCH}} + \text{InnerOffset}_{\text{DRCH}} + b) \bmod (N_{\text{SUBZONE}}(s) / N_{\text{BLOCK}}), N_{\text{DRCH-USABLE}} / N_{\text{BLOCK}}, \text{ where PBRI}(i, M) \text{ is generated as specified in 9.2.5.2.}$$

- e) Compute INDEX_{AVAIL-DRCH}(DRCH, q, s, b, r) = (ZoneOffset_{DRCH} + N_{MIN-DRCH-SPACING} × RefPos_{DRCH} + min(RefPos_{DRCH}, N_{MAX-DRCH-SPACINGS}) + r × (N_{DRCH-AVAIL-SUBCARRIERS} / N_{BLOCK})) mod N_{DRCH-AVAIL-SUBCARRIERS}.

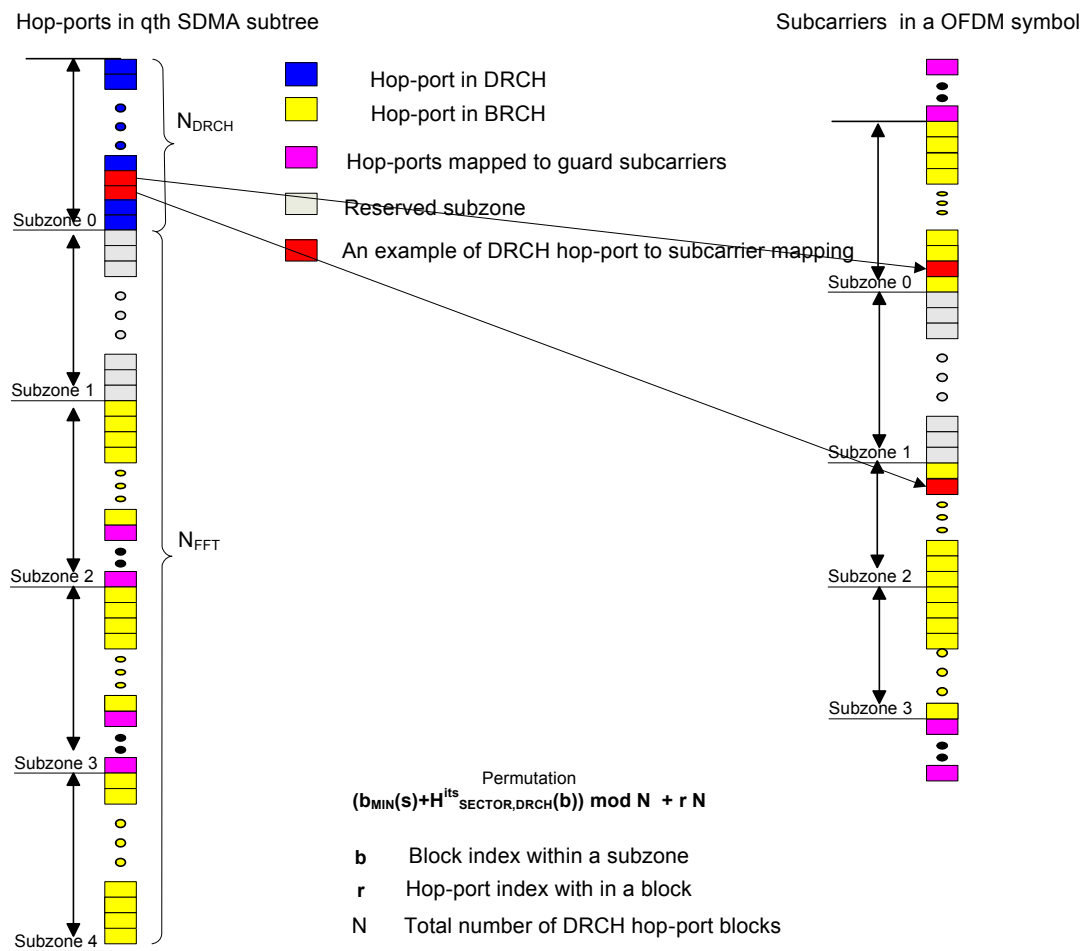


Figure 100—Illustration of DRCH hop-port to subcarrier mapping if
ResourceChannelMuxMode = 1

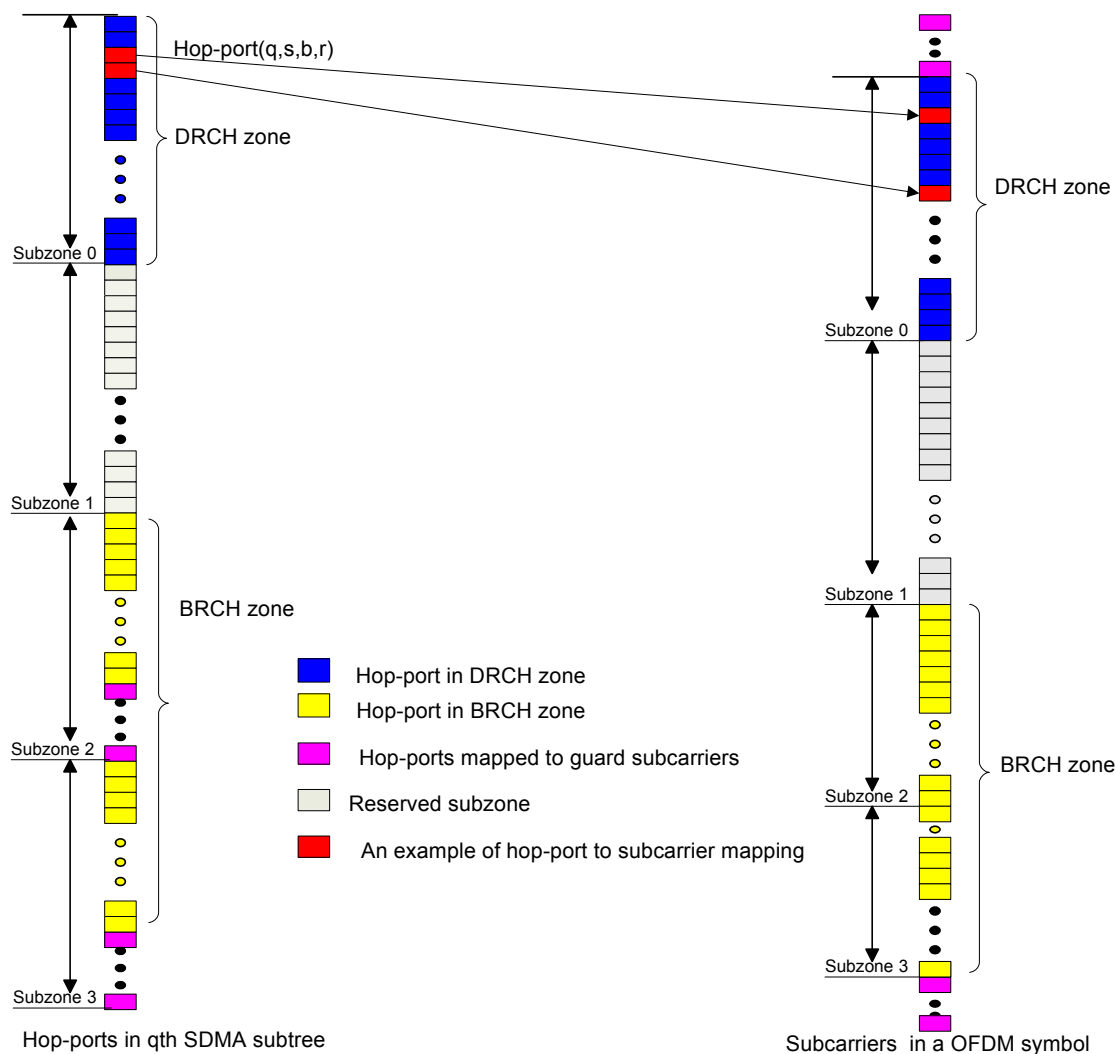


**Figure 101—Illustration of DRCH hop-port to subcarrier mapping if
ResourceChannelMuxMode = 2**

9.2.14.6 Forward Link Control Segment hop-ports and exchanged hop-ports

Every hop-port allocated to the Forward Link Control Segment is associated with an exchanged hop-port in every Forward Link PHY frame.²⁷ The hop-ports allocated to the Forward Link Control Segment and the corresponding exchanged hop-ports are defined in 9.2.15.1. Let F_{MAPPING} denote the hop-port to subcarrier

²⁷ Allocation procedure aims at approximately equal number of usable hop-port blocks borrowed from every usable subzone of the respective zone. This goal is achieved by populating Forward Link Control Segment hop-port blocks over usable subzones in a round robin fashion. Each hop-port block is further associated with a subcarrier block of size N_{BLOCK} . Subcarrier allocation rule secures at least third order diversity with respect to channel fading and other sector interference when the Forward Link Control Segment is allocated in the BRCH zone and higher order otherwise. This is accomplished by dividing the set of all usable subcarriers arranged in the physical order into three disjoint control hopping zones of approximately equal size and mapping three consecutive hop-port blocks of the Forward Link Control Segment to three different control hopping zones when UseDRCHForFLCS = 0 and ResourceChannelMuxMode = 2, as illustrated by Figure 99. Since hop-port to subcarrier mapping of BRCH resources defined in 9.2.14.5.1 maps contiguous sets of hop-port blocks onto contiguous subcarrier blocks, this can be achieved in the hop-port space, by exchanging hop-port blocks assigned to Forward Link Control Segment with other hop-port blocks within BRCH zone. Exchanging hop-port blocks rather than subcarrier blocks makes the following procedure applicable to all the cases of Forward Link Control Segment allocation, specifically to the case of UseDRCHForFLCS field being equal to '1' if ResourceChannelMuxMode = 1 and to the cases of UseDRCHForFLCS field equal to '0' or '1' if ResourceChannelMuxMode = 2.

mapping as described in 9.2.14.5.1 for BRCH hop-ports and in 9.2.14.5.2 for DRCH hop-ports. For all hop-ports allocated to the Forward Link Control Segment,

- Let p_{FLCS} be the hop-port allocated to the Forward Link Control Segment and let p_{EXCH} be the corresponding exchanged hop-port.
- The hop-port p_{FLCS} shall be mapped to the subcarrier $F_{MAPPING}(p_{EXCH})$.
- The hop-port p_{EXCH} shall be mapped to the subcarrier $F_{MAPPING}(p_{FLCS})$.

9.2.15 Forward Link Control Segment resource allocation and indexing

In every PHY frame in a superframe, the Forward Link control channels, i.e., Forward Acknowledgment Channel, Forward Start-of-Packet Channel, Forward Shared Control Channel, Forward Fast Other Sector Interference Channel, Forward Interference over Thermal Channel, Forward Pilot Quality Indicator Channel and Forward Power Control Channel shall be multiplexed together onto a set of $N_{FLCS-BLOCKS}$ hop-port blocks, referred to as the 'Forward Link Control Segment.' The notion of hop-port block is defined in 9.2.14.4. The value of $N_{FLCS-BLOCKS}$ is given by $N_{FLCS-COMMON-BLOCKS} + 3 \times N_{FLCS-LAB-SEGMENTS}$, where $N_{FLCS-COMMON-BLOCKS}$ and $N_{FLCS-LAB-SEGMENTS}$ are given by the NumCommonSegmentHopPortBlocks and NumLABSegments fields of the Overhead Messages Protocol (see 11.6) respectively. The hop-port blocks or tiles shall be indexed 0 through $N_{FLCS-BLOCKS} - 1$. The hop-ports in these blocks shall be referred to as "Forward Link Control Segment hop-ports." The tiles shall be referred to as "Forward Link Control Segment tiles." The Forward Link Control Segment shall operate on only DRCH resources if ResourceChannelMuxMode = 1 and can operate on either only the DRCH resources or only the BRCH resources if ResourceChannelMuxMode = 2. The choice of BRCH or DRCH is specified by the UseDRCHForFLCS field of Overhead Messages Protocol.

9.2.15.1 Allocation of hop-port blocks and subcarrier blocks to the Forward Link Control Segment

A set of $N_{FLCS-BLOCKS}$ hop-port blocks of size N_{BLOCK} is allocated to the Forward Link Control Segment within each Forward Link PHY frame. These hop-port blocks shall be located within the DRCH zone if the value of UseDRCHForFLCS field is '1' and shall be located within the BRCH zone otherwise.

The following procedure enumerates all the usable hop-port blocks within the zone where Forward Link Control Segment is allocated.

- a) Initialize a hop-port block counter b to 0. Initialize a counter k , of usable hop-port blocks, to 0.
- b) If hop-port block b in SDMA subtree 0 comprises of only usable hop-ports as defined in 9.2.14.4.4 one of the following conditions holds:
 - 1) UseDRCHForFLCS field equals '1' and b is part of the DRCH zone;
 - 2) UseDRCHForFLCS field equals '0' and b is part of the BRCH zone.

then set $FLCSUsableBlock[k] = b$ and increment k by 1.
- c) Increment b by 1.
- d) Repeat steps b) and c) until one of the following conditions hold:
 - 1) UseDRCHForFLCS field equals '1' and DRCH hop-port blocks are exhausted.
 - 2) UseDRCHForFLCS field equals '0' and BRCH hop-port blocks are exhausted.
- e) Set TotalNumBlocks = k .

Allocation of hop-port blocks to the Forward Link Control Segment is defined as follows:

- a) Initialize a tile counter k of Forward Link Control Segment hop-port blocks to 0. Initialize a subzone counter s to 0. Initialize S counters $b_0, b_1, \dots, b_{s-1}$ of hop-port blocks within subzones to 0.

- b) If $b_s < N_{\text{SUBZONE}}(s) / N_{\text{BLOCK}}$ and one of the following conditions holds:
 - 1) UseDRCHForFLCS field equals '1', subzone indexed s is part of the DRCH zone and b_s is a usable hop-port block within this subzone;
 - 2) UseDRCHForFLCS field equals '0', subzone indexed s is part of the BRCH zone and b_s is a usable hop-port block within this subzone;

then

 - 3) Define the k^{th} hop-port block FLCSHopPortBlock[k] of the Forward Link Control Segment to be a block of N_{BLOCK} consecutive hop-ports (BRCH, 0, s , b_s , 0) through (BRCH, 0, s , b_s , $N_{\text{BLOCK}} - 1$) if UseDRCHForFLCS field equals '0' and to a block of N_{BLOCK} consecutive hop-ports (DRCH, 0, s , b_s , 0) through (DRCH, 0, s , b_s , $N_{\text{BLOCK}} - 1$) if UseDRCHForFLCS field equals '1'.
 - 4) Increment b_s by 1.
 - 5) Increment k by 1.
- c) Set s to $(s + 1) \bmod S$.
- d) If $k < N_{\text{FLCS-BLOCKS}}$ then repeat steps b) and c) of this list.

Define the association of exchanged hop-port blocks to the hop-port blocks allocated to the Forward Link Control Segment. The set of usable hop-port blocks is divided into three control hopping zones of sizes $M_0 = \text{TotalNumBlocks}/3$, $M_1 = \text{TotalNumBlocks}/3$ if $\text{TotalNumBlocks} \bmod 3 = 2$ and $M_1 = \text{TotalNumBlocks}/3$ otherwise, and $M_2 = \text{TotalNumBlocks}/3$. Intra-zone permutations H_0^{ij} , H_1^{ij} , and H_2^{ij} of sizes M_0 , M_1 , and M_2 corresponding to the Forward Link PHY frame indexed j and superframe indexed i are defined as follows:

- Set $\text{SEED}_k = f_{\text{PHY-HASH}}(15 \times 2^{10} \times 32 \times 4 + \text{PilotID} \times 32 \times 4 + (i \bmod 32) \times 4 + k)$ where SectorSeed is as defined in 9.2.2.2.3 and $f_{\text{PHY-HASH}}$ is the hash function described in 9.2.5.4.
- H_k^i is the permutation of size M_k generated with seed SEED_k using the common permutation generation algorithm described in 9.2.5.1, $0 \leq k < 3$. Note that H_k^i is independent of the Forward Link PHY frame index and therefore is constant over a superframe.
- H_k^{ij} as a m^{th} order cyclic shift of the permutation H_k^i : $H_k^{ij}(n) = H_k^i((n + m) \bmod M_k)$, $0 \leq n < M_k$, where $m = ((f_{\text{PHY-HASH}}(\text{PilotID} + j + 1))) \bmod M_k$.

Association of exchanged hop-port blocks to the hop-port blocks of the Forward Link Control Segment in the Forward Link PHY frame indexed j and superframe indexed i is performed according to the following procedure:

- a) Initialize the counter k of Forward Link Control Segment hop-port blocks to 0. Initialize the counter m of exchanged hop-ports to 0. Initialize three counters c_0 , c_1 , c_2 of usable hop-port blocks within each of the three control hopping zones to 0.
- b) Set $d = m \bmod 3$.
- c) If $c_d < M_d$ then
 - 1) Set the exchanged hop-port block ExchHopPortBlock $_{ij}[k]$ associated with the k^{th} hop-port block FLCSHopPortBlock[k] of the Forward Link Control Segment to be the $(D + H_d^{ij}(c_d))^{\text{th}}$ usable hop-port block FLCSUsableBlock[$D + H_d^{ij}(c_d)$] where $D = 0$ if $d = 0$, $D = M_0$ if $d = 1$, and $D = (M_0 + M_1)$ if $d = 2$.
 - 2) Increment c_d by 1
 - 3) Increment m by 1
 - 4) Increment k by 1
 - 5) Proceed to step c)

otherwise

- 4) Increment m by 1
- 5) Repeat step b) and step c)
- d) If $k < N_{\text{FLCS-BLOCKS}}$ then repeat steps b) and c).

When the k^{th} hop-port block $\text{FLCSHopPortBlock}[k]$ of the Forward Link Control Segment is exchanged with the $\text{ExchHopPortBlock}_{ij}[k]$ hop-port block in the Forward Link PHY frame indexed j and superframe indexed i , the subcarrier block corresponding to the hop-port block $\text{FLCSHopPortBlock}[k]$ shall be mapped by the hop-port block $\text{ExchHopPortBlock}_{ij}[k]$ while the subcarrier block corresponding to the hop-port block $\text{ExchHopPortBlock}_{ij}[k]$ shall be mapped by the hop-port block $\text{FLCSHopPortBlock}[k]$. Specifically, let $p_0, p_1, \dots, p_{N_{\text{BLOCK}}-1}$ be a set of contiguous hop-ports within the hop-port block $\text{FLCSHopPortBlock}[k]$ and let $p'_0, p'_1, \dots, p'_{N_{\text{BLOCK}}-1}$ be a set of contiguous hop-ports within the hop-port block $\text{ExchHopPortBlock}_{ij}[k]$. In the Forward Link PHY frame indexed j and superframe indexed i and OFDM symbol indexed t , the m^{th} hop-port within the hop-port block $\text{FLCSHopPortBlock}[k]$ shall map to the subcarrier mapped by the hop-port p'_m according to the mapping algorithm defined in 9.2.14.5.1 for BRCH hop-ports and 9.2.14.5.2 for DRCH hop-ports for $0 \leq m < N_{\text{BLOCK}}$. Similarly, the m^{th} hop-port within the hop-port block $\text{ExchHopPortBlock}_{ij}[k]$ shall map to the subcarrier mapped by the hop-port p_m according to the mapping algorithm defined in 9.2.14.5.1 for BRCH hop-ports and 9.2.14.5.2 for DRCH hop-ports for $0 \leq m < N_{\text{BLOCK}}$.

Note that allocation of hop-port blocks to the Forward Link Control Segment is static while allocation of the associated exchanged hop-port blocks depends on the Forward Link PHY frame index and the superframe index, and is sector specific.

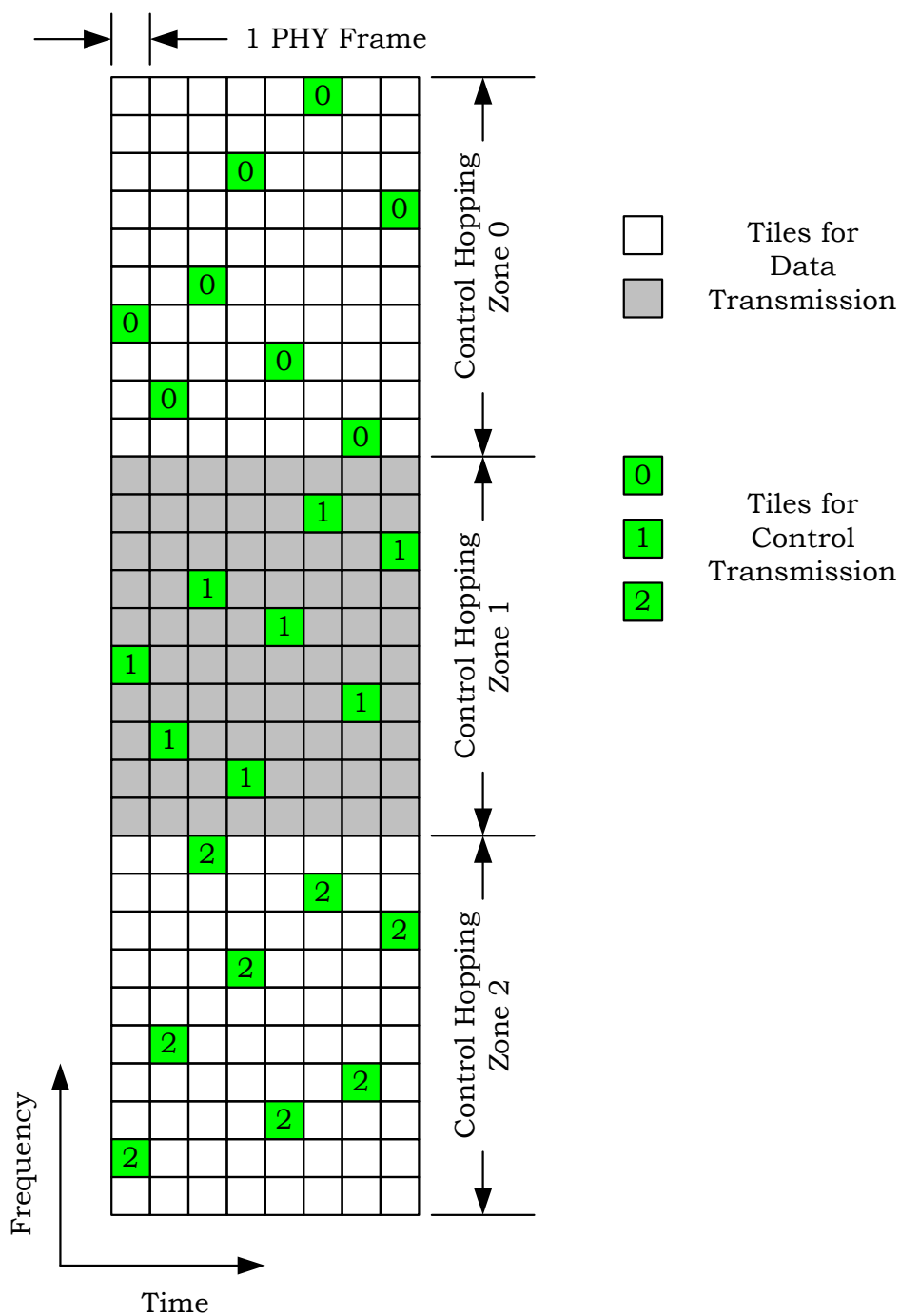


Figure 102—Illustration of Forward Link Control Segment hopping with BRCH resources

9.2.15.2 Forward Link Control Segment resource definition and indexing

Since the Forward Link control channels, i.e., the Forward Acknowledgment Channel, Forward Start-of-Packet Channel, the Forward Reverse Activity Bit Channel, the Forward Fast Other Sector Interference Channel, the Forward Interference over Thermal Channel, and the Forward Pilot Quality Indicator Channel

are transmitted using third-order diversity, the Forward Link Control Segment hop-ports are divided into groups of three, each of which is referred to as a “Forward Link Control Segment resource.” Different types of channels shall not co-exist on any given Forward Link Control Segment. For example, if the Forward Acknowledgment Channel is transmitted on one hop-port in a Forward Link Control Segment resource, then the Forward Acknowledgment Channel shall be transmitted on the other two hop-ports as well.

The Forward Link Control Segment resources are indexed as $R_{FLCS} = \{0, 1, 2, \dots\}$ and are modulated starting with Forward Acknowledgment Channel, followed by the Forward Start-of-Packet Channel, the Forward Reverse Activity Bit Channel, the Forward Pilot Quality Indicator Channel, the Forward Fast Other Sector Interference Channel, Forward Interference over Thermal Channel, and the Forward Power Control Channel. Details of the Forward Link Control Segment resources used for each such channel is in the description of the respective channel.

9.2.15.2.1 Forward Link Control Segment partitioning

The Forward Link Control Segment is partitioned into the Common Segment (CS) and zero or more LAB Segments (LS). The term LAB refers to Link Assignment Block as defined in 8.5.

The Common Segment shall be present in all Forward Link PHY frames and shall contain all the Common Forward Control Channels, where the Common Forward Control Channels consist of all channels carried on the Forward Link Control Segment except the Forward Shared Control Channel. These channels include the Forward Acknowledgment Channel, Forward Start-of-Packet Channel, Forward Fast Other Sector Interference Channel, Forward Interference over Thermal Channel, Forward Pilot Quality Indicator Channel and Forward Power Control Channel. The Common Segment shall contain Forward Link Control Segment blocks with indices 0 through $N_{FLCS-COMMON-BLOCKS} - 1$, where $N_{FLCS-COMMON-BLOCKS}$ is equal to the NumCommonSegmentHopPortBlocks field of the Overhead Messages Protocol.

The LAB Segments, if present, shall contain only the Forward Shared Control Channel. The LAB Segment with index j shall contain the Forward Link Control Segment Blocks with indices $N_{FLCS-COMMON-BLOCKS} + 3j$ through $N_{FLCS-COMMON-BLOCKS} + 3j + 2$.

In addition to the Common Forward Control Channels, the Common Segment shall contain the Forward Shared Control Channel if one of the following conditions are satisfied:

- UseDRCHForFLCS = 1 (i.e., when FLCS is transmitted on the DRCH).
- UseDRCHForFLCS = 0 and no LAB Segments are present. In this case, the Common Segment shall not contain the Forward Shared Control Channel if $N_{FLCS-COMMON-BLOCKS} > 9$ (i.e., the access network shall not set NumLABSegments to 0 if NumCommonSegmentHopPortBlocks is greater than 9).

9.2.15.2.2 Forward Link Control Segment resource indexing

The Forward Link Control Segment resources shall only map to hop-ports in the Common Segment. The allocation of subcarrier-symbols to Forward Link Control Segment resources depends on the UseDRCHForFLCS field of the Overhead Messages Protocol.

9.2.15.2.2.1 Forward Link Control Segment resource indexing for UseDRCHForFLCS = 0

Any three hop-port blocks with consecutive indices allocated to the Forward Link Control Segment are located in three different control hopping zones (see 9.2.15.1). To guarantee third-order diversity for all the Common Forward Control Channels except the Forward Power Control Channel, three subcarrier-symbols of each Forward Link Control Segment resource shall be placed in three hop-port blocks with consecutive hop-port block indices referred to as TileSegments, and are indexed as $\{0, 1, 2\}$.

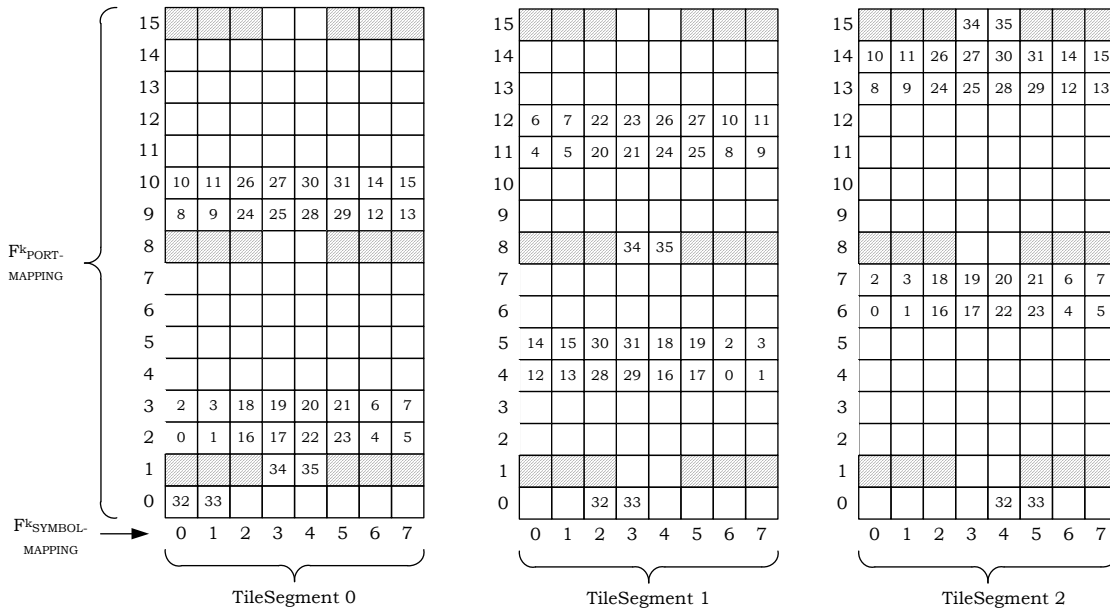


Figure 103—TileSegments for Forward Link Control Segment

The structure of TileSegments is illustrated in Figure 103. Since the total number of subcarrier-symbols available for control channels in every hop-port block is 110, there can be at most $\lfloor 110/3 \rfloor \times N_{\text{FLCS-COMMON-BLOCKS}}$ Forward Link Control Segment resources in the Common Segment. Hence, the total number of Forward Link Control Segment resources is $36 \times N_{\text{FLCS-COMMON-BLOCKS}}$.

The Forward Link Control Segment resources shall be populated in groups of four such that most groups of four consecutive Forward Link Control Segment resources map to a block of two subcarriers over two OFDM symbols. This ensures that the mapping of the Forward Acknowledgement Channel and the Forward Start-of-Packet Channel to a group of four consecutive Forward Link Control Segment resources results in these channels occupying two subcarriers over two OFDM symbols.

A Forward Link Control Segment resource with index R_{FLCS} with the minimum value of 0 and the maximum value of $36 \times N_{\text{FLCS-COMMON-BLOCKS}}$ shall be allocated three subcarrier symbols according to the following procedure:

- a) Define $h_{\text{BLOCK-INDEX}} = (\lfloor R_{\text{FLCS}}/4 \rfloor) \bmod N_{\text{FLCS-COMMON-BLOCKS}}$.
- b) Define $r_{\text{INTRA-INDEX}} = 4 \times \lfloor R_{\text{FLCS}} / (4N_{\text{FLCS-COMMON-BLOCKS}}) \rfloor + (R_{\text{FLCS}} \bmod 4)$.
- c) For $k = \{0, 1, 2\}$
 - 1) Let $p_k = F^k_{\text{PORT-MAPPING}}(r_{\text{INTRA-INDEX}})$ and $t_k = F^k_{\text{SYMBOL-MAPPING}}(r_{\text{INTRA-INDEX}})$, where $F^k_{\text{PORT-MAPPING}}$ and $F^k_{\text{SYMBOL-MAPPING}}$ are the port and symbol mappings for TileSegment k as shown in Figure 103.
 - 2) Let $h_k = (h_{\text{BLOCK-INDEX}} \times 4 + k) \bmod N_{\text{FLCS-COMMON-BLOCKS}}$.
 - 3) The hop-port with index p_k in Orthogonal Frequency Division Multiplexing symbol with index t_k in Forward Link Control Segment tile with index h_k shall be allocated to the Forward Link Control Segment resource R_{FLCS} .

9.2.15.2.2.2 Forward Link Control Segment resource indexing when UseDRCHForFLCS = 1

The Forward Link Control Segment resources are allocated to prevent collisions with the Forward Common Pilot Channels and the Forward Beacon Pilot Channel. The number of available subcarrier-symbols in a given tile shall be computed according to the following procedure:

- a) Initialize a port counter i , an OFDM symbol counter j and an available modulation symbol counter k to 0.
- b) If hop-port i in any of the hop-port blocks 0 through $N_{\text{FLCS-COMMON-BLOCKS}}$ is mapped to a pilot subcarrier, then the subcarrier-symbol corresponding to the hop-port i in the OFDM symbol j shall be unavailable.

Otherwise:

- 1) The subcarrier symbol is available and is assigned the index k .
- 2) Increment k by 1.
- c) Increment i by 1. If $i = N_{\text{BLOCK}}$, set $i = 0$ and increment j by 1.
- d) If $j = N_{\text{FRAME}}$, set $N_{\text{AVAILABLE}}$ to k and halt the procedure. Otherwise, go to step b).

The Forward Link Control Segment resource with index R_{FLCS} shall be allocated three subcarrier-symbols according to the following procedure:

- Set $h_{\text{TILE}} = (R_{\text{FLCS}} \bmod N_{\text{FLCS-COMMON-BLOCKS}})$ and r_{INTRA} to floor $(R_{\text{FLCS}} / N_{\text{FLCS-COMMON-BLOCKS}})$.
- For $k = 0, 1, 2, \dots$, the k^{th} subcarrier-symbol of the Forward Link Control Segment resource shall be assigned the available index $k \lfloor N_{\text{AVAILABLE}}/3 \rfloor + r_{\text{INTRA}}$ in the Forward Link Control Segment tile with index $(h_{\text{TILE}} + k) \bmod N_{\text{FLCS-COMMON-BLOCKS}}$. The sector shall ensure that r_{INTRA} is always less than $\lfloor N_{\text{AVAILABLE}}/3 \rfloor$.

9.3 Requirements for access terminal operation

This subclause defines requirements that are specific to access terminal equipment and operation. An access terminal may support operation in one or more band classes.

9.3.1 Transmitter

9.3.1.1 Frequency parameters

9.3.1.1.1 Channel spacing and designation

See 3GPP2 C.S0057-B for a description of the band classes that an access terminal may support.

9.3.1.1.2 Frequency tolerance

The access terminal shall meet the requirements of 9.3.1.5.1.

9.3.1.2 Power output characteristics

All power levels are referenced to the access terminal antenna connector unless otherwise specified.

9.3.1.2.1 Maximum output power

When implementing operation for one or more band classes as specified in 3GPP2 C.S0057, the access terminal shall meet the requirements of 3GPP2 C.S0011-C.

The access terminal shall be capable of transmitting at the minimum specified power level when transmitting only on the Access Channel, the Reverse CDMA Dedicated Control Channel, the Reverse CDMA Data Channel, the Reverse OFDMA Dedicated Control Channel, or the Reverse OFDMA Data Channel, and when commanded to maximum output power. The output power may be lower when transmitting on more than one of the following: the Reverse CDMA Dedicated Control Channel, Reverse CDMA Data Channel, Reverse OFDMA Dedicated Control Channel, Reverse OFDMA Data Channel, or the Reverse Acknowledgment Channel. The access terminal shall not exceed the maximum specified power levels under any circumstances.

9.3.1.2.2 Output power limits

9.3.1.2.2.1 Minimum controlled output power

When implementing operation for one or more band classes as specified in 3GPP2 C.S0057, the access terminal shall meet the requirements of 3GPP2 C.S0011-C.

9.3.1.3 Modulation characteristics

9.3.1.3.1 Reverse Link signals

Each of the PHY frames is further divided into a Code Division Multiple Access segment and an Orthogonal Frequency Division Multiple Access segment. Subclauses 9.3.1.3.3 and 9.3.1.3.4 list the set of channels that are part of the Code Division Multiple Access segment and the Orthogonal Frequency Division Multiple Access segment, respectively.

9.3.1.3.1.1 Channel structures

The structure of the code channels transmitted by a access terminal is shown in Table 212.

Table 212—Reverse Link channels

Pilot channels	
R-PICH	Reverse Pilot Channel
R-AuxPICH	Reverse Auxiliary Pilot Channel
R-DPICH	Reverse Dedicated Pilot Channel

Control channels	
R-ACH	Reverse Access Channel
R-CDCH	Reverse Code Division Multiple Access Dedicated Control Channel
R-ODCH	Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel
R-ACKCH	Reverse Acknowledgment Channel

Traffic channels	
R-CDCH	Reverse Code Division Multiple Access Data Channel
R-ODCH	Reverse Orthogonal Frequency Division Multiple Access Data Channel

9.3.1.3.1.1.1 Reverse Link OFDMA channels

The Reverse Link OFDMA channels consist of the Reverse Dedicated Pilot Channel, the Reverse OFDMA Dedicated Control Channel, the Reverse Acknowledgement Channel, and the Reverse OFDMA Data Channel.

The structure of the Reverse Acknowledgment Channel is shown in Figure 104. The structure of the Reverse OFDMA Dedicated Control Channel is shown in Figure 105. The structure of the Reverse OFDMA Data Channel is shown in Figure 106.

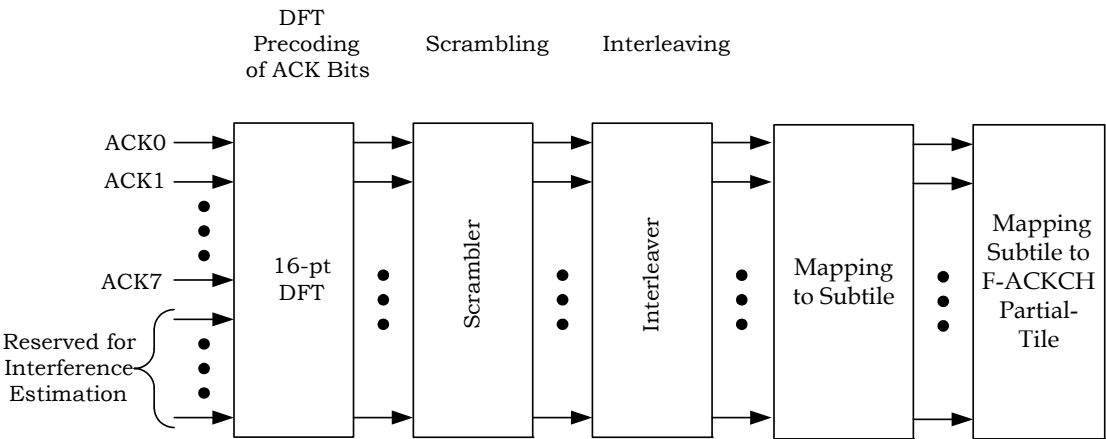


Figure 104—Channel structure for Reverse Acknowledgment Channel

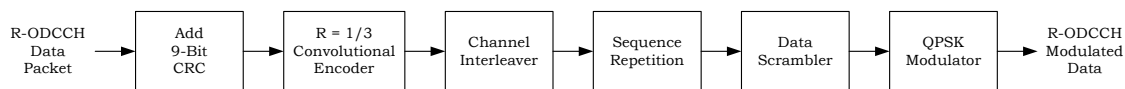


Figure 105—Channel structure for Reverse OFDMA Dedicated Control Channel

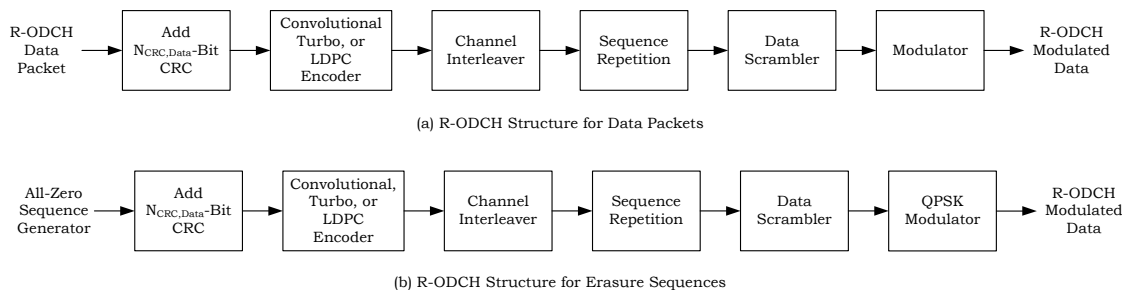


Figure 106—Channel structure for Reverse OFDMA Data Channel

9.3.1.3.1.1.2 Reverse Link CDMA channels

The Reverse Link CDMA channels consist of the Reverse Pilot Channel, the Reverse Auxiliary Pilot Channel, the Reverse Access Channel, the Reverse CDMA Dedicated Control Channel, and the Reverse CDMA Data Channel.

The CDMA segment is generated as shown in Figure 107. The CDMA and OFDMA segments are multiplexed as shown in Figure 108.

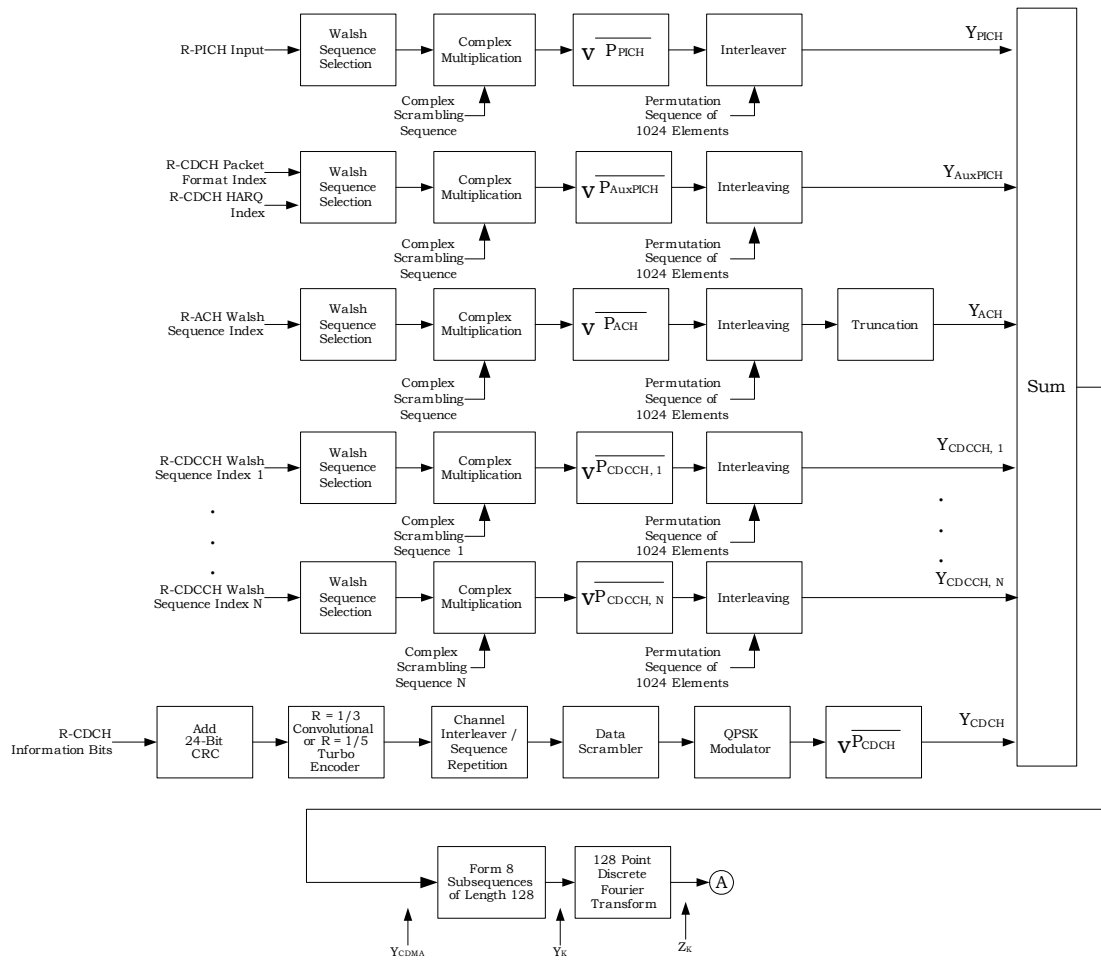


Figure 107—Structure of the Reverse Link Code Division Multiple Access Segment for the i^{th} Code Division Multiple Access Subsegment

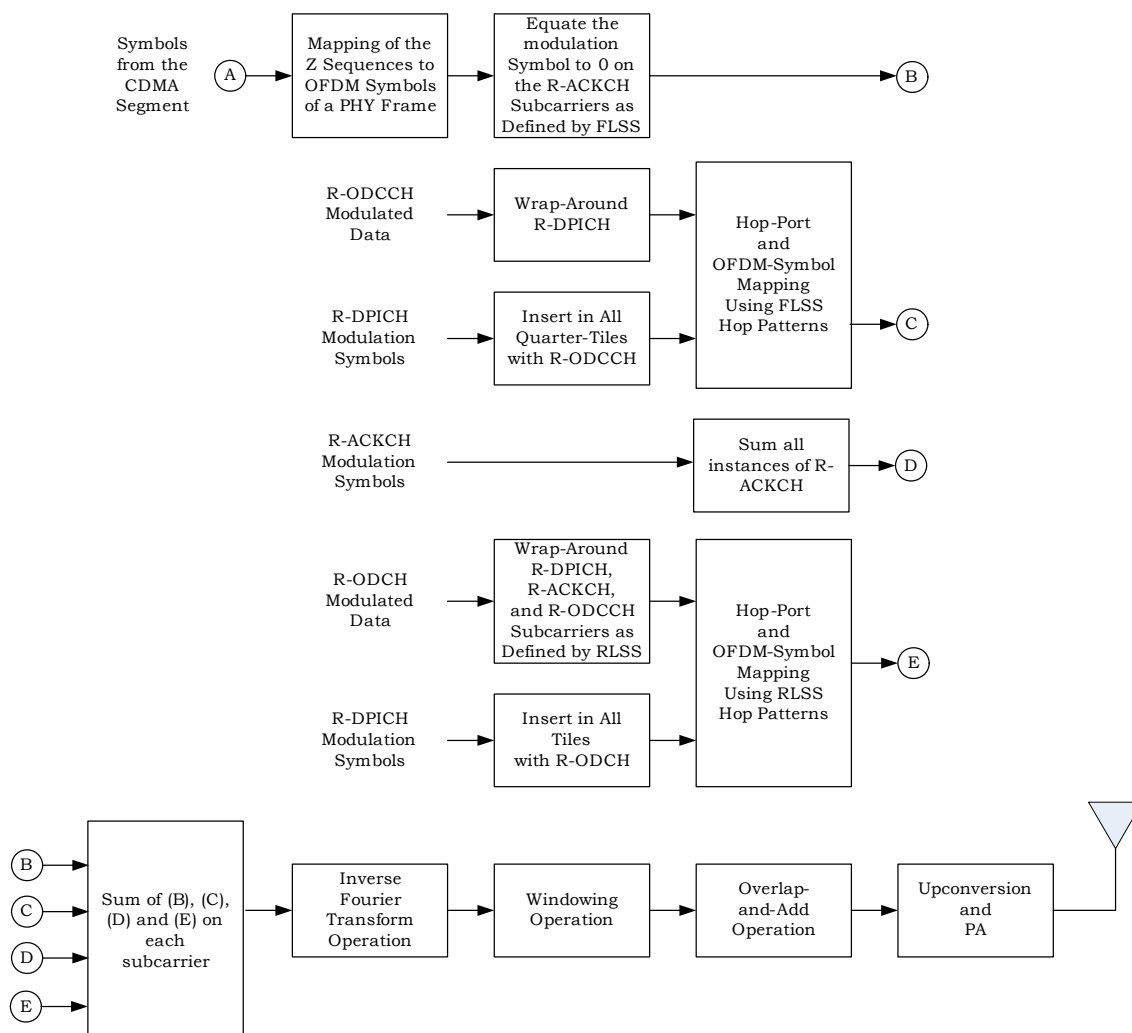


Figure 108—Structure of the Reverse Link Orthogonal Frequency Division Multiple Access Segment and the Code Division Multiple Access/Orthogonal Frequency Division Multiple Access Multiplexing

9.3.1.3.2 Code Division Multiple Access structure and modulation parameters

The Code Division Multiple Access segment carries the Reverse Access Channel, the Reverse Pilot Channel and the Reverse Code Division Multiple Access Dedicated Control Channel. The CDMA segment can also carry the Reverse Code Division Multiple Access Data Channel and the Reverse Code Division Multiple Access Auxiliary Pilot Channel. Transmissions from different access terminals in the Code Division Multiple Access segment are multiplexed in a Code Division Multiple Access fashion, i.e., they are not orthogonal with respect to each other.

The waveforms corresponding to the different channels that are carried on the Code Division Multiple Access segment are first generated in the time-domain. The time-domain waveforms of the different channels are then added together and the resulting waveform is converted to the frequency domain using a Discrete Fourier Transform (DFT) operation. The resulting frequency-domain sequence is then mapped to

the subcarriers of an Orthogonal Frequency Division Multiplexing symbol that are assigned to the Code Division Multiple Access segment for this access terminal.

The Code Division Multiple Access segment for each sector may consist of one or more Code Division Multiple Access subsegments in an Reverse Link PHY frame. The number of Code Division Multiple Access subsegments, as well as the subcarriers assigned to each of them, is specified in 9.2.10. The Code Division Multiple Access segment modulation, including the set of subcarriers on which the frequency-domain waveform is modulated, depends on the target sector of that channel. For each Reverse Link PHY frame, the Lower MAC Layer may instruct the Physical Layer to transmit one or more channels on the Code Division Multiple Access segment, and these channels may be targeted to the same or to different sectors. For each channel, the Lower MAC layer also specifies the subsegment (in the case of all channels except the Reverse Code Division Multiple Access Data Channel) or the set of subsegments (in the case of the Reverse Code Division Multiple Access Data Channel) on which the channel is to be transmitted, in terms of a subsegment index or a set of subsegment indices. Note that Code Division Multiple Access segment transmissions to one sector may overlap with Orthogonal Frequency Division Multiple Access segment transmissions to other sectors, in which case also the two shall be superimposed.

The set of subcarriers corresponding to each subsegment index for a given sector in a given Reverse Link PHY frame is determined by the hop permutation, and is defined in 9.2.10.

9.3.1.3.2.1 Time-interleaving of the Code Division Multiple Access channels

The procedures described in this subclause are carried out separately for each channel in each Code Division Multiple Access subsegment. For the Reverse Code Division Multiple Access Data Channel, these procedures are carried out separately on each subsequence that is transmitted on a separate Code Division Multiple Access subsegment.

A permutation H_{CTRL} of size 1024 shall be generated using the common permutation algorithm in 9.2.5.1 using the seed [0100 0011 0000 0101 0100].

A sequence X of length 1024 shall be time-interleaved to generate a sequence Y of length 1024 according to $Y(i) = X(H_{CTRL}(i))$, where the notation $A(i)$ denotes the i^{th} element of the sequence A .

9.3.1.3.2.2 Multiplexing the Code Division Multiple Access channels

The procedures described in this subclause shall be carried out separately for each Code Division Multiple Access subsegment.

For each Code Division Multiple Access subsegment in a Reverse Link PHY frame, the time-domain sequences corresponding to all the channels to be transmitted in that subsegment shall be added together to form a time-domain sequence Y_{CDMA} . Note that for the Reverse Code Division Multiple Access Data Channel, only the subsequence to be transmitted in the subsegment of interest is part of the addition.

The Reverse Code Division Multiple Access Data Channel may be transmitted over multiple subsegments in any given PHY frame, the set of subsegments being determined by the RCC MAC Protocol. The time domain Reverse Code Division Multiple Access Data Channel sequence (X_{CDCH}) shall be split into multiple sequences, each of length 1024, each of which is transmitted over a single subsegment. The elements indexed 0 through 1023 of X_{CDCH} form the first subsequence $X_{CDCH,0}$, the elements indexed 1024 to 2047 form the second subsequence $X_{CDCH,1}$, etc. The sequence $X_{CDCH,i}$ shall be transmitted in the i^{th} subsegment allocated to the Reverse Code Division Multiple Access Data Channel in any given PHY frame.

9.3.1.3.2.3 Discrete Fourier Transform (DFT) operation

The procedures described in this subclause shall be carried out independently for each Code Division

Multiple Access subsegment in the Reverse Link PHY frame.

The interleaved time-domain sequence Y_{CDMA} for each Code Division Multiple Access subsegment generated in the previous subclause shall be broken up into $N_{\text{FRAME}} = 8$ different subsequences of length 128. The first 128 elements of the sequence Y_{CDMA} form the first subsequence Y_0 , the next 128 elements form the second sequence Y_1 , etc. Each of these subsequences shall be converted to a frequency domain sequence through a Discrete Fourier Transform (DFT) operation, which is defined in 9.2.5.5.

Let Z_k denote the DFT of the 128-length subsequence Y_k , $0 \leq k \leq 7$. For all Reverse Link PHY frames except the Reverse Link PHY frame with index 0 in a superframe, the sequence Z_k shall be modulated onto the Orthogonal Frequency Division Multiplexing symbol with index k in the Reverse Link PHY frame. For the Reverse Link PHY frame with index 0 in a superframe, the sequence Z_k shall be modulated onto the Orthogonal Frequency Division Multiplexing symbol with index $k + 8$.

The elements of the sequence Z_k shall be modulated onto the subcarriers allocated to the Code Division Multiple Access segment in the designated Orthogonal Frequency Division Multiplexing symbol (k or $k + 8$, as described above) in increasing order, i.e., the element $Z_k(0)$ shall be modulated onto the subcarrier with the lowest index in the Code Division Multiple Access subsegment, the element $Z_k(1)$ shall be modulated onto the subcarrier with the next-lowest index, and the element $Z_k(127)$ shall be modulated onto the subcarrier with the highest index in the Code Division Multiple Access subsegment.

After mapping of the sequence Z_k on the subcarriers in the Code Division Multiple Access subsegment, the modulation symbols corresponding to the subcarriers allocated to the Reverse Acknowledgment Channel of the FLSS, or the subcarriers corresponding to the Reverse Link silence interval of the RLSS shall be set to 0. The modulation symbols corresponding to these subcarriers shall be blanked, i.e., transmitted with zero energy.

9.3.1.3.3 Code Division Multiple Access Segment

The Code Division Multiple Access segment carries the Reverse Access Channel, the Reverse Pilot Channel, the Reverse Auxiliary Pilot Channel, and Reverse Code Division Multiple Access Dedicated Control Channel. Additionally, it can also carry the Reverse Code Division Multiple Access Data Channel.

9.3.1.3.3.1 Reverse Pilot Channel

The Reverse Pilot Channel (R-PICH) is an unmodulated DFT-precoded Code Division Multiple Access signal used to assist the access network for Reverse Link power control reference and Reverse Link quality measurement. The access terminal shall transmit the Reverse Pilot Channel on CDMA subsegments assigned by the RLSS, the FLSS, and all the sectors that are mutually synchronous with the RLSS, subject to the capability of the access terminal on the maximum supportable Reverse Pilot Channel subsegments per frame, as determined by the RCC MAC Protocol (see 8.7). Transmission on subsegments assigned by the RLSS take priority over subsegments assigned by other sectors; subsegments assigned by the FLSS take priority over non-serving sectors. The access terminal shall determine the timing alignment used to transmit the Reverse Pilot Channel on each CDMA subsegment in accordance with the following rules:

- a) The access terminal shall transmit at most one instance of the Reverse Pilot Channel on any CDMA subsegment.
- b) The access terminal shall use the same timing alignment to transmit all instances of the Reverse Pilot Channel on the subsegments that overlap with one another (i.e., over at least one subcarrier-symbol).
- c) The access terminal shall use the same timing alignment on all the Reverse Pilot Channel subsegments assigned by the same sector.
- d) The access terminal shall use the access terminal timing reference of the RLSS to time-align its Reverse Pilot Channel transmissions on subsegments assigned by the RLSS.

- e) All transmissions to the RLSS shall be time-aligned with the access terminal time-base of the RLSS. When the access terminal transmits the Reverse Pilot Channel on multiple subsegments in the same frame, the timing alignment used to transmit the Reverse Pilot Channel (as determined by the above rules) may not coincide with the access terminal time-base reference for some of the non-RLSS sectors which assigned those CDMA subsegments to the access terminal. Nevertheless, the access terminal may use its time-base reference for those sectors to transmit other control channels (e.g., the Reverse CDMA Dedicated Control Channel) to those sectors.

9.3.1.3.3.1.1 Reverse Pilot Channel modulation

The Reverse Pilot Channel time-domain sequence shall be the Walsh sequence W_0^{1024} , where the notion of a Walsh sequence is as defined in 9.2.5.6.

9.3.1.3.3.1.2 Reverse Pilot Channel scrambling

The Reverse Pilot Channel time-domain sequence shall be multiplied element-wise with a complex scrambling sequence of length 1024 and scaled by the quantity $\sqrt{P_{\text{PICH}}}$, where P_{PICH} is the power density determined by the RCC MAC Protocol at which the Reverse Pilot Channel is to be transmitted. The scrambling sequence is as defined in 9.2.5.2 with the seed determined by the following procedure:

- a) Let X denote the 49-bit quantity $\text{RPICHScramblingSeed}$, which is public data of the FLCS MAC Protocol. The $\text{RPICHScramblingSeed}$ identifies the access terminal to all its active set members.
- b) Let Y denote the 9-bit Superframe Index and Z a 6-bit representation of the Reverse Link PHY frame Index within the superframe.
- c) Concatenate the quantities X , Y , and Z , with X forming the LSBs and Z the MSBs to obtain a 64 bit quantity W .
- d) The seed for the Reverse Pilot Channel scrambling sequence shall be $f_{\text{PHY-HASH}}(W)$, where the common PHY hash-function $f_{\text{PHY-HASH}}$ is defined in 9.2.5.4.

9.3.1.3.3.1.3 Reverse Pilot Channel time-interleaving

The scrambled Reverse Pilot Channel sequence shall be time-interleaved according to the procedure described in 9.3.1.3.2.1.

9.3.1.3.3.1.4 Reverse Pilot Channel multiplexing

The scrambled sequence of the Reverse Pilot Channel shall be multiplexed to form a Code Division Multiple Access time domain sequence as described in 9.3.1.3.2.2.

9.3.1.3.3.1.5 Reverse Pilot Channel DFT operation

The DFT operation on the Code Division Multiple Access subsegment containing the Reverse Pilot Channel transmission shall be performed as described in 9.3.1.3.2.3.

9.3.1.3.3.2 Reverse Auxiliary Pilot Channel

The Reverse Auxiliary Pilot Channel (R-AuxPICH) is transmitted in every subsegment containing an Reverse Code Division Multiple Access Data Channel transmission. It is used to assist the access network in decoding an access terminal transmission on the Reverse Code Division Multiple Access Data Channel. In addition, this channel also carries information about the rate and HARQ transmission index of the Reverse Code Division Multiple Access Data Channel transmission in the same subsegment.

Transmission on the Reverse Auxiliary Pilot Channel is aligned with the transmission on the Reverse Code Division Multiple Access Data Channel.

9.3.1.3.3.2.1 Reverse Auxiliary Pilot Channel modulation

The Reverse Auxiliary Pilot Channel time-domain sequence shall be the Walsh sequence W_i^{1024} , where $i = \text{PFID} \times 6 + \text{HARQTransmissionIndex}$, where PFID denotes the packet format index used for the transmission of the Reverse Code Division Multiple Access Data Channel in the same subsegment, and is determined by the RTC MAC Protocol. HARQTransmissionIndex denotes the HARQ transmission index for the Reverse Code Division Multiple Access Data Channel in the same subsegment, and is also determined by the RTC MAC Protocol.

9.3.1.3.3.2.2 Reverse Auxiliary Pilot Channel scrambling

The Reverse Auxiliary Pilot Channel time-domain sequence shall be multiplied element-wise with a complex scrambling sequence of length 1024 and scaled by the quantity $\sqrt{P_{\text{AuxPICH},k}}$, where $P_{\text{AuxPICH},k}$ is the power per modulation symbol of the PHY frame k at which the Reverse Auxiliary Pilot Channel is to be transmitted. P_{AuxPICH} is determined by the RTC MAC Protocol 8.6. The scrambling sequence is as defined in 9.2.5.2 with the seed given by the output of the hash function defined in 9.2.5.4 with input equal to $(32 \times 2048 \times p + 32 \times m + 6)$, where p is the SectorSeed corresponding to the target sector and is as defined in 9.2.2.2.3 and m is the MACID of the terminal in the target sector.

9.3.1.3.3.2.3 Reverse Auxiliary Pilot Channel time-interleaving

The scrambled Reverse Auxiliary Pilot Channel sequence shall be time-interleaved according to the procedure described in 9.3.1.3.2.1.

9.3.1.3.3.2.4 Reverse Auxiliary Pilot Channel multiplexing

The scrambled sequence of the Reverse Auxiliary Pilot Channel shall be multiplexed to form a Code Division Multiple Access time domain sequence as described in 9.3.1.3.2.2.

9.3.1.3.3.2.5 Reverse Auxiliary Pilot Channel DFT operation

The DFT operation on the Code Division Multiple Access subsegment containing the Reverse Auxiliary Pilot Channel transmission shall be performed as described in 9.3.1.3.2.3.

9.3.1.3.3.3 Reverse Access Channel

The Reverse Access Channel (R-ACH) is used by the access terminal for initial access, for transition out of semi-connected state, or to hand off between sectors at the same or at different frequencies.

The AC MAC instructs the Physical Layer to transmit on the Reverse Access Channel in a particular Code Division Multiple Access subsegment. Along with this instruction, the AC MAC also provides a WalshSequenceID, an AccessScramblingID, and a probe transmission power.

9.3.1.3.3.3.1 Reverse Access Channel modulation

The Reverse Access Channel time-domain sequence shall be the Walsh sequence $W_{\text{WalshSequenceID}}^{1024}$, where the notion of a Walsh sequence is defined in 9.2.5.6.

9.3.1.3.3.3.2 Reverse Access Channel scrambling

The Reverse Access Channel time-domain sequence shall be multiplied element-wise with a complex scrambling sequence of length 1024 and scaled by the quantity $\sqrt{P_{\text{ACH}}}$, where P_{ACH} is the access probe power determined by the RCC MAC Protocol (see 8.7) at which the Reverse Access Channel is to be transmitted. The scrambling sequence is defined in 9.2.5.2 with the seed given by the output of the hash

function defined in 9.2.5.4 with input equal to $32 \times p + (\text{AccessScramblingID} \bmod 16) \times 2 + 1$, where p is the SectorSeed corresponding to the target sector and is defined in 9.2.2.2.3.

9.3.1.3.3.3.3 Reverse Access Channel time-interleaving

The scrambled Reverse Access Channel sequence is time-interleaved according to the procedure described in 9.3.1.3.2.1.

9.3.1.3.3.3.4 Reverse Access Channel truncation

The last 128 entries of the time-interleaved Reverse Access Channel sequence, i.e., the elements with indices 896 through 1023, shall be set to zero.

9.3.1.3.3.3.5 Reverse Access Channel multiplexing

The truncated sequence of the Reverse Access Channel shall be multiplexed to form a Code Division Multiple Access time domain sequence as described in 9.3.1.3.2.2.

9.3.1.3.3.3.6 Reverse Access Channel DFT operation

The DFT operation on the Code Division Multiple Access subsegment containing the Reverse Access Channel transmission shall be performed as described in 9.3.1.3.2.3.

9.3.1.3.3.4 Reverse Code Division Multiple Access Dedicated Control Channel

The Reverse Code Division Multiple Access Dedicated Control Channel (R-CDCCCH) is a Code Division Multiple Access channel that can carry one or more of the following logical channels: the channel quality indicator channel, the request channel, the power amplifier headroom channel, and the power spectral density indication channel.

The RCC MAC Protocol instructs the Physical Layer to transmit one or more instances of the Reverse Code Division Multiple Access Dedicated Control Channel in a given Code Division Multiple Access subsegment. The Lower MAC Layer provides a 10-bit input data that shall be interpreted as a Walsh sequence index (WalshSequenceID). The Lower MAC Layer also provides a transmit power P_{CDCCCH} .

9.3.1.3.3.4.1 Reverse Code Division Multiple Access Dedicated Control Channel modulation

The Reverse Code Division Multiple Access Dedicated Control Channel time-domain sequence on the Reverse Link shall be the Walsh sequence $W_{\text{WalshSequenceID}}^{1024}$, where the notion of a Walsh sequence is defined in 9.2.5.6.

9.3.1.3.3.4.2 Reverse Code Division Multiple Access Dedicated Control Channel scrambling

The Reverse Code Division Multiple Access Dedicated Control Channel time-domain sequence shall be multiplied element-wise with a complex scrambling sequence of length 1024 and scaled by the quantity $\sqrt{P_{\text{CDCCCH}}}$, where P_{CDCCCH} is the power density determined by the RCC MAC at which the Reverse Code Division Multiple Access Dedicated Control Channel is to be transmitted. The scrambling sequence is defined in 9.2.5.2 with the seed given by $f_{\text{PHY-HASH}}(32 \times 2048 \times p + 32 \times m + 4 \times (s \bmod 8) + 3)$, where $f_{\text{PHY-HASH}}$ is defined in 9.2.5.4, p is the SectorSeed corresponding to the target sector defined in 9.2.2.2.3, m is the MACID of the terminal in the target sector, and s is the SHOGID. If no SHOG_ID is specified, s shall be set to 0.

9.3.1.3.3.4.3 Reverse Code Division Multiple Access Dedicated Control Channel time-interleaving

The scrambled Reverse Code Division Multiple Access Dedicated Control Channel sequence shall then be time-interleaved according to the procedure described in 9.3.1.3.2.1.

9.3.1.3.3.4.4 Reverse Code Division Multiple Access Dedicated Control Channel multiplexing

The scrambled sequence of the Reverse Code Division Multiple Access Dedicated Control Channel shall be multiplexed to form a Code Division Multiple Access time domain sequence as described in 9.3.1.3.2.2.

9.3.1.3.3.4.5 Reverse Code Division Multiple Access Dedicated Control Channel DFT operation

The DFT operation on Code Division Multiple Access subsegment containing the Reverse Code Division Multiple Access Dedicated Control Channel transmission shall be performed as described in 9.3.1.3.2.3.

9.3.1.3.3.5 Reverse Code Division Multiple Access Data Channel

The Reverse CDMA Data Channel (R-CDCH) may be used for the transmission of higher-level data to the access networks by the access terminals.

9.3.1.3.3.5.1 Reverse Code Division Multiple Access Data Channel encoding

Each Reverse CDMA Data Channel packet is generated by the RTC MAC Protocol (see 8.8), and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled, and QPSK modulated according to the procedure described in 9.2.6.1. A CRC length of $N_{\text{CRC,DATA}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(7 \times 2048 \times \text{SectorSeed} + m \bmod 2048)$ shall be used for the data scrambling operation.

9.3.1.3.3.5.2 Reverse Code Division Multiple Access Data Channel multiplexing

The multiplexing operation on Code Division Multiple Access segment containing the Reverse Code Division Multiple Access Data Channel data shall be performed as described in 9.3.1.3.2.2.

9.3.1.3.3.5.3 Reverse Code Division Multiple Access Data Channel DFT operation

The DFT operation on the Code Division Multiple Access segment carrying the Reverse Code Division Multiple Access Data Channel data shall be performed as described in 9.3.1.3.2.3.

9.3.1.3.4 Orthogonal Frequency Division Multiple Access segment

The Orthogonal Frequency Division Multiple Access segment is comprised of the Reverse Dedicated Pilot Channel, Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel, Reverse Acknowledgment Channel, and the Reverse Orthogonal Frequency Division Multiple Access Data Channel.

9.3.1.3.4.1 Reverse Dedicated Pilot Channel

The Reverse Dedicated Pilot Channel (R-DPICH) is used to provide dedicated pilots for the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel and Reverse Orthogonal Frequency Division Multiple Access Data Channel in order to allow an Access Point to perform channel estimation.

As described in 9.2.10, the hop-ports on the Reverse Link are divided into units of hop-port blocks. Each hop-port block consists of 16 contiguous hop-ports, which are mapped by the hopping permutation to a

contiguous set of subcarriers. Also, the set of subcarriers corresponding to a hop-port block does not change over one PHY frame. Therefore, the set of resources (over time and frequency) can be divided into units of tiles, where a tile is a contiguous 16x8 rectangle of hop-ports (16 in frequency and 8 in time), which are mapped to a contiguous 16x8 rectangle of subcarriers (16 in frequency and 8 in time).

Each tile on the Reverse Link can be assigned to the Code Division Multiple Access segment, to the Orthogonal Frequency Division Multiple Access segment or may be left blank. The Reverse Dedicated Pilot Channel shall be present only in tiles assigned to the Orthogonal Frequency Division Multiple Access segment. With the Orthogonal Frequency Division Multiple Access segment, the tile may be assigned to the Reverse Orthogonal Frequency Division Multiple Access Data Channel or the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel. A Reverse Acknowledgment Channel partial-tile may hop onto a Reverse Orthogonal Frequency Division Multiple Access Data Channel tile (as described in 9.2.11). The Reverse Dedicated Pilot Channel pilot patterns are different for each of the cases mentioned above.

Some of the subcarriers in each tile shall be designated as Reverse Dedicated Pilot Channel subcarriers. The Reverse Dedicated Pilot Channel configuration in each tile depends on the following parameters:

- a) Reverse Dedicated Pilot Channel format: For the Reverse Orthogonal Frequency Division Multiple Access Data Channel tiles, the Reverse Dedicated Pilot Channel format can take one of two values, indexed 0 and 1. For tiles occupied by the Reverse Orthogonal Frequency Division Multiple Access Data Channel, the Reverse Dedicated Pilot Channel format depends on the Reverse Orthogonal Frequency Division Multiple Access Data Channel assignment occupying this tile, and is determined by the RTC MAC Protocol (see 8.8).
- b) Energy per modulation symbol: This quantity is denoted by P . The value of P for a tile assigned to the Reverse Orthogonal Frequency Division Multiple Access Data Channel shall be equal to P_{ODCH} , where P_{ODCH} is the power density of that tile as specified by the RTC MAC Protocol (see 8.8). The value of P for all tiles assigned to the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel shall be equal to P_{ODCCH} , where P_{ODCCH} is the power density for the Reverse OFDMA Dedicated Control Channel specified by the RCC MAC Protocol (see 8.7).
- c) CodeOffset: This is an integer between 0 and 2. For tiles belonging to the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel, the CodeOffset shall be equal to 0. For tiles belonging to the Reverse Orthogonal Frequency Division Multiple Access Data Channel, the value is determined by the value of SubtreeIndex for that Reverse Orthogonal Frequency Division Multiple Access Data Channel assignment, which is determined by the RTC MAC Protocol. CodeOffset is equal to zero if SubtreeIndex is equal to zero. For other values of SubtreeIndex, the value of CodeOffset is given by $\text{RLDPICHCodeOffset}_{\text{SubtreeIndex}j}$, which is a field of the Overhead Messages Protocol.

9.3.1.3.4.1.1 Reverse Dedicated Pilot Channel for Reverse Orthogonal Frequency Division Multiple Access Data Channel Tiles

The locations of the Reverse Dedicated Pilot Channel subcarriers in a tile depend on the Reverse Dedicated Pilot Channel format and are shown in Figure 109. Note that the hop-ports within a tile are indexed 0 to 15 in increasing order of hop-port index, and the Orthogonal Frequency Division Multiplexing symbols within a Reverse Link PHY frame are indexed 0 to 7 with the earliest Orthogonal Frequency Division Multiplexing symbol being indexed 0.

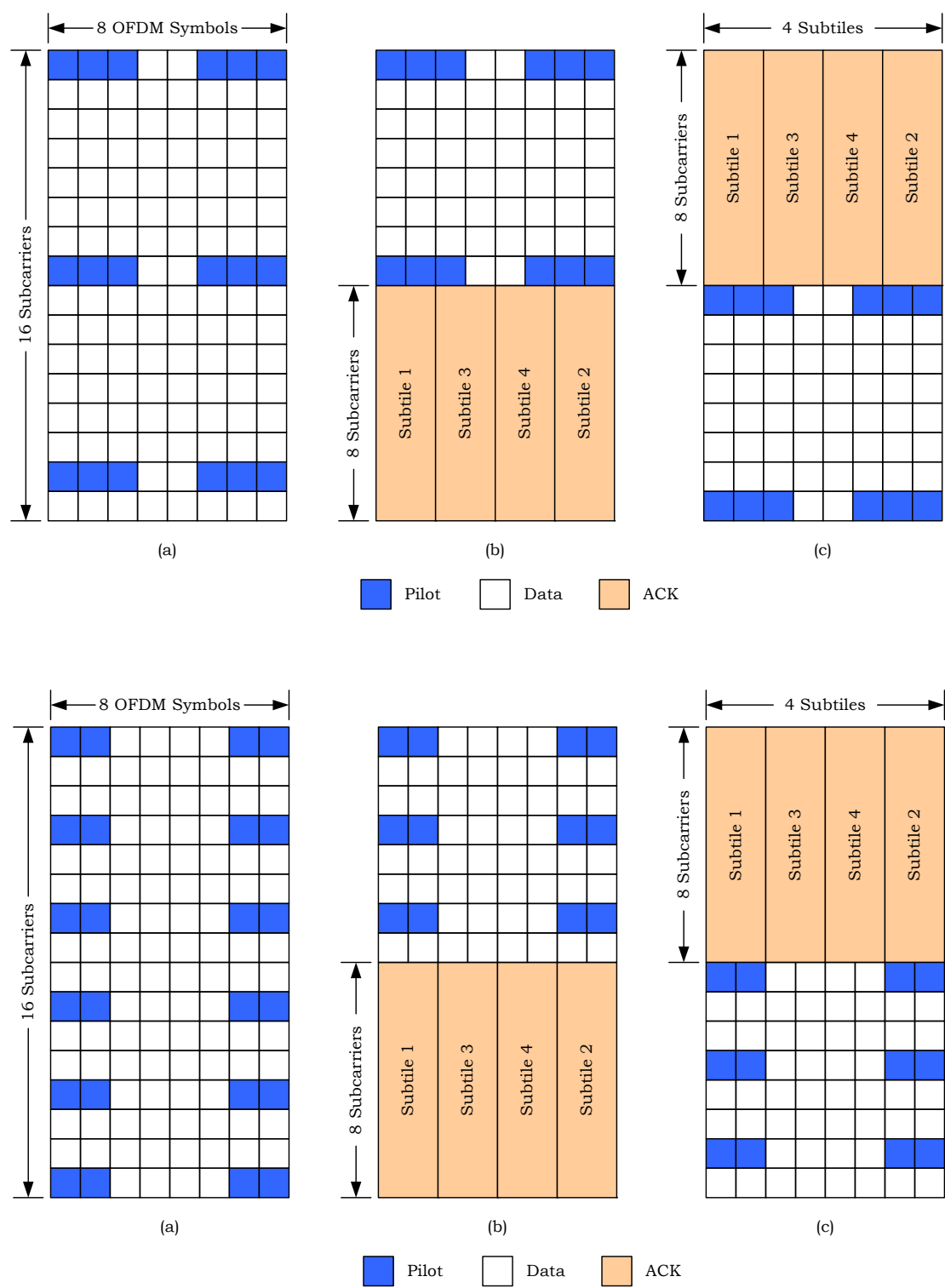


Figure 109—Location of Reverse Dedicated Pilot Channel subcarriers within a tile for the Different Reverse Dedicated Pilot Channel formats

9.3.1.3.4.1.1 Reverse Dedicated Pilot Channel Pilot Formats 0 and 1

For the Reverse Dedicated Pilot Channel Format 0, the Reverse Dedicated Pilot Channel shall occupy the subcarrier symbol of the tile if the Orthogonal Frequency Division Multiplexing symbol index t within the Reverse Link PHY frame is in the set

- $\{0, 1, 2, 5, 6, 7, 8, 9, 10, 13, 14, 15\}$ for Reverse Link PHY frame 0 when DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 0.
- $\{8, 9, 10, 13, 14, 15\}$ for Reverse Link PHY frame 0 when DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 0 and Reverse Link silence interval transmission coincides with Reverse Link PHY frame 0.
- $\{0, 1, 2, 5, 6, 7\}$ for all other cases.

and if the hop-port index within the tile is in the set

- $\{1, 8, 15\}$ when no Reverse Acknowledgment Channel partial-tile is mapped to any of the subcarriers in the Reverse OFDMA Data Channel tile.
- $\{1, 8, 15\}$ when a Reverse Acknowledgment Channel partial-tile is mapped to the same subcarriers as hop-ports 0 through 7 within the tile for some OFDM symbol.
- $\{0, 7, 14\}$ when a Reverse Acknowledgment Channel partial-tile is mapped to the same subcarriers as hop-ports 8 through 15 within the tile for some OFDM symbol.

and if the subcarrier-symbol is not occupied by the Reverse Acknowledgement Channel, where the Reverse Acknowledgment Channel partial-tile allocation is as described in 9.2.11.

The complex value of all Reverse Dedicated Pilot Channel modulation symbols in Orthogonal Frequency Division Multiplexing symbol indexed t shall be given by

$$S_t = \sqrt{P_{\text{ODCH}}} \exp \left(\frac{j2\pi}{3} (\text{CodeOffset})(t \bmod 8) \right) \text{ if } t \bmod 8 < 4, \text{ and}$$

$$S_t = \sqrt{P_{\text{ODCH}}} \exp \left(\frac{j2\pi}{3} (\text{CodeOffset})(7 - (t \bmod 8)) \right) \text{ (if } t \bmod 8 \geq 4),$$

where j denotes the complex number (0, 1), and P_{ODCH} denotes the energy per modulation symbol of the Reverse OFDMA Data Channel in the tile.

For Reverse Dedicated Pilot Channel Format 1, the Reverse Dedicated Pilot Channel shall occupy the modulation symbol of the tile if the Orthogonal Frequency Division Multiplexing symbol index t is in the set

- $\{0, 1, 6, 7, 8, 9, 14, 15\}$ for Reverse Link PHY frame 0 when DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 0.
- $\{8, 9, 14, 15\}$ for Reverse Link PHY frame 0 when DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 0 and Reverse Link silence interval transmission coincides with Reverse Link PHY frame 0.
- $\{0, 1, 6, 7\}$ in all other cases.

the hop-port index within the tile is in the set $\{0, 3, 6, 9, 12, 15\}$, and if the subcarrier-symbol is not occupied by the Reverse Acknowledgment Channel. The process of partial-tile allocation for the Reverse Acknowledgment Channel is described in 9.2.11.

The complex value of all Reverse Dedicated Pilot Channel modulation symbols in Orthogonal Frequency Division Multiplexing symbol indexed t shall be given by

$$S_t = \sqrt{P_{\text{ODCH}}} \exp(j\pi(\text{CodeOffset})(t \bmod 8)) \text{ if } t \bmod 8 < 4, \text{ and}$$

$$S_t = \sqrt{P_{\text{ODCH}}} \exp(j\pi(\text{CodeOffset})(7 - (t \bmod 8))) \text{ (if } t \bmod 8 \geq 4).$$

9.3.1.3.4.1.1.2 Reverse Dedicated Pilot Channel scrambling

Scrambling for the Reverse Dedicated Pilot Channel is done on a tile-by-tile basis for the Reverse OFDMA Data Channel. In the case of an extended Reverse Link PHY frame, i.e., a Reverse Link PHY frame with index 0 in the superframe when DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 0, each block of 16 subcarriers shall consist of two tiles, with the first set of N_{FRAME} OFDM symbols forming the first tile and the second set of N_{FRAME} OFDM symbols forming the second tile. The Reverse Dedicated Pilot Channel indexing and scrambling operations shall be carried out independently for the two tiles. In this case, the Reverse Dedicated Pilot Channel index definitions (9.3.1.3.4.1.1.2.1) and scrambling sequences (9.3.1.3.4.1.1.2.2) are computed independently for the first eight OFDM symbols and the last eight OFDM symbols of that Reverse Link PHY frame.

9.3.1.3.4.1.1.2.1 Reverse Dedicated Pilot Channel index definition

The scrambling symbols shall be generated only for subcarriers that correspond to Reverse Dedicated Pilot Channel hop-ports (via the hop-permutation), as defined in 9.3.1.3.3.4.1. These subcarriers are henceforth referred to as Reverse Dedicated Pilot Channel subcarriers. For the purpose of scrambling, the Reverse Dedicated Pilot Channel subcarriers in each tile shall be indexed by a quantity called the Reverse Dedicated Pilot Channel index. The Reverse Dedicated Pilot Channel index shall be computed according to the following procedure for the Reverse Orthogonal Frequency Division Multiple Access Data Channel tiles:

- a) Initialize an Orthogonal Frequency Division Multiplexing symbol counter i , a subcarrier counter j and an Reverse Dedicated Pilot Channel index counter k to 0.
- b) If the subcarrier j in Orthogonal Frequency Division Multiplexing symbol i within the tile is an Reverse Dedicated Pilot Channel subcarrier, then
 - 1) Set its Reverse Dedicated Pilot Channel index to k .
 - 2) Increment k by 1.
- c) Increment i by 1. If $i = N_{\text{FRAME}}$, set i to 0 and increment j .
- d) Repeat steps b) and c) until $j = N_{\text{BLOCK}}$.

Thus, the Reverse Dedicated Pilot Channel subcarriers are indexed in time first, followed by frequency.

9.3.1.3.4.1.1.2.2 Scrambling sequence

The scrambling symbols for a tile depend on the tile index T , which shall be equal to $(f_{\text{MIN}} - N_{\text{GUARD, LEFT}}) / N_{\text{BLOCK}}$, where f_{MIN} is the lowest indexed subcarrier in that tile. For the tile with index T within any PHY frame in the superframe with index SFInd, a complex scrambling sequence shall be generated using the common complex scrambling algorithm described in 9.2.5.2 with seed $f_{\text{PHY-HASH}}[34 \times 2^{20} \times 8 + \text{SectorSeed} \times 8 + T \bmod 8]$, where SectorSeed is defined in 9.2.2.2.3. The k^{th} symbol $c(k)$ in the complex scrambling sequence shall be used to scramble the Reverse Dedicated Pilot Channel subcarrier with the Reverse Dedicated Pilot Channel index k . The scrambling operation shall consist of multiplying the unscrambled complex symbol on the subcarrier with the scrambling symbol $c(k)$.

9.3.1.3.4.1.2 Reverse Dedicated Pilot Channel for Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel quarter-tiles

The Reverse Dedicated Pilot Channel is present in each quarter-tile allocated to the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel. The description and allocation of quarter-

tiles are as described in 9.3.1.3.4.1.1. Unlike the Reverse Orthogonal Frequency Division Multiple Access Data Channel pilot formats, the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel pilot format does not change with Reverse Acknowledgment Channel hopping. This is because the Reverse Acknowledgment Channel half-tiles do not hop onto the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel quarter-tiles.

9.3.1.3.4.1.2.1 Reverse Dedicated Pilot Channel Pilot Pattern for Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel

The locations of the Reverse Dedicated Pilot Channel subcarriers in a quarter-tile are shown in Figure 110. Note that the subcarriers within a quarter-tile are indexed 0 to 7 in increasing order of subcarriers index, and the Orthogonal Frequency Division Multiplexing symbols within the quarter-tile are indexed 0 to 3 with the earliest Orthogonal Frequency Division Multiplexing symbol being indexed 0.

The Reverse Dedicated Pilot Channel shall occupy a subcarrier symbol if the subcarrier index within the quarter-tile is in the set $\{0, 7\}$ and the Orthogonal Frequency Division Multiplexing symbol index t within the quarter-tile is in the set $\{0, 1, 2, 3\}$. The complex value of the Reverse Dedicated Pilot Channel modulation symbol shall be given by $S_t = \sqrt{P_{\text{ODCCH}}}$.

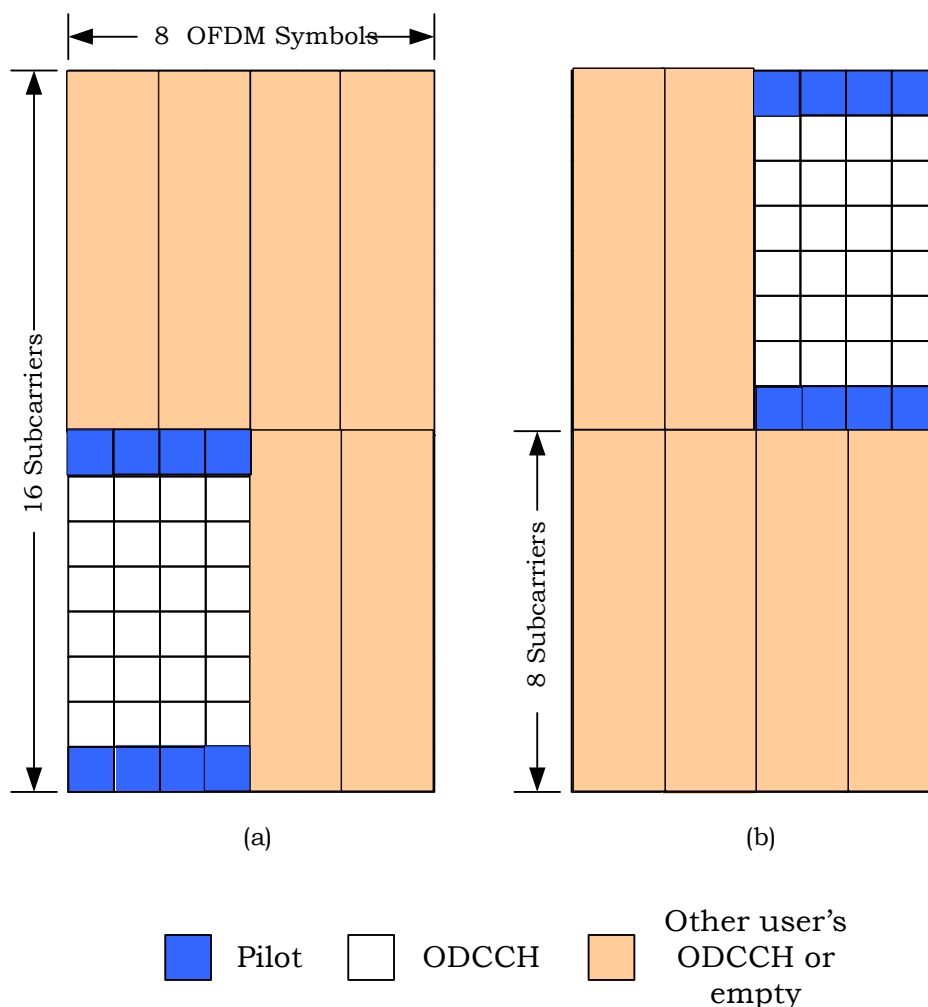


Figure 110—Location of Reverse Dedicated Pilot Channel subcarriers within a reverse OFDMA Dedicated Channel quarter-tile

9.3.1.3.4.1.2.2 Reverse Dedicated Pilot Channel scrambling

9.3.1.3.4.1.2.2.1 Reverse Dedicated Pilot Channel index definition

Reverse Dedicated Pilot Channel scrambling is done on a quarter-tile basis for the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel. The scrambling symbols that shall be used shall only be generated for Reverse Dedicated Pilot Channel subcarriers. For the purpose of scrambling, the Reverse Dedicated Pilot Channel subcarriers in quarter-tile shall be indexed by a quantity called the Reverse Dedicated Pilot Channel index. The Reverse Dedicated Pilot Channel index shall be computed according to the following procedure for the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel quarter-tiles:

- a) Initialize an Orthogonal Frequency Division Multiplexing symbol counter i , a subcarrier counter j and a Reverse Dedicated Pilot Channel index counter k to 0.
- b) If the subcarrier j in Orthogonal Frequency Division Multiplexing symbol i within the quarter-tile is a Reverse Dedicated Pilot Channel subcarrier, then
 - 1) Set its Reverse Dedicated Pilot Channel index to k .
 - 2) Increment k by 1.
- c) Increment i by 1. If $i = N_{\text{FRAME}}/2$ set i to 0 and increment j .
- d) Repeat steps b) and c) until $j = N_{\text{BLOCK}}/2$.

Thus, the Reverse Dedicated Pilot Channel subcarriers are indexed in time first, followed by frequency.

9.3.1.3.4.1.2.2.2 Scrambling sequence

The scrambling symbols for a quarter-tile depend on the RODCResourceIndex of that quarter-tile. The RODCResourceIndex is as described in 9.3.1.3.4.1. For the tile with index T within any PHY frame in the superframe with index SF_{Ind} , a complex scrambling sequence shall be generated using the common complex scrambling algorithm described in 8.7.5.2 with seed $f_{\text{PHY-HASH}}[37 \times 2^{20} \times 32 + \text{SectorSeed} \times 32 + \text{RODCResourceIndex} \bmod 32]$, where SectorSeed is defined in 9.2.2.2.3. The k^{th} symbol $c(k)$ in the complex scrambling sequence shall be used to scramble the Reverse Dedicated Pilot Channel subcarrier with the Reverse Dedicated Pilot Channel indexed k . The scrambling operation shall consist of multiplying the unscrambled complex symbol on the subcarrier with the scrambling symbol $c(k)$.

9.3.1.3.4.2 Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel

The Reverse OFDMA Dedicated Control Channel payload multiplexes several logical channels, namely the $r\text{-cqich}$, the $r\text{-reqch}$, the $r\text{-mqich}$, the $r\text{-sfch}$, and the $r\text{-bfch}$. The RCC MAC Protocol (see 8.7) may instruct the Physical Layer to transmit one or more Reverse OFDMA Dedicated Control Channel instances in a Reverse Link PHY frame.

9.3.1.3.4.2.1 Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel resource assignment

The access terminal shall be allocated two quarter-tiles as a function of RODCResourceIndex, which is a field of the RCC MAC Protocol (see 8.7). Let $t_A = \lfloor \text{RODCResourceIndex}/2 \rfloor$ and $k_A = (\text{RODCResourceIndex} \bmod 2)$. The two quarter-tiles assigned shall be those with indices (t_A, k_A) and $((t_A + N_{\text{TILES}}/2) \bmod N_{\text{TILES}}, k_A + 2)$, where N_{TILES} and quarter-tiles are defined in 9.2.12.

9.3.1.3.4.2.2 Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel modulation

The Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel modulation consists of transmitting a packet generated by the RCC MAC Protocol (see 8.7) on the two quarter tiles assigned to the access terminal. These two quarter-tiles are determined as a function of RODCResourceIndex as specified in 9.3.1.3.4.2.

The Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel packet generated by the RCC MAC Protocol is appended with CRC, encoded using the rate-1/3 convolutional code, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.3.1.3.4.4. A CRC length of $N_{\text{CRC,ODCCH}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(2^{31} \times (s \bmod 8) + 2048 \times \text{SectorSeed} + m \bmod 2048)$ shall be used for the data scrambling operation, where the SectorSeed corresponds to the FLSS and is defined in 9.2.2.2.3; m is the MACID of the access terminal as associated with the FLSS provided by the RCC MAC Protocol (see 8.7), and s is the SHOGID specified by the RCC MAC Protocol (see 8.7). If no SHOGID is specified, then s is set to 0.

The Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel packet shall be modulated on the two quarter-tiles as follows:

- a) Initialize a quarter-tile counter q , Orthogonal Frequency Division Multiplexing symbol counter t and subcarrier counter f to 0.
- b) If the subcarrier f in Orthogonal Frequency Division Multiplexing symbol t within quarter tile q is not a Reverse Dedicated Pilot Channel subcarrier and is not part of the Reverse Link Silence Interval, then a QPSK modulation symbol s is generated by the modulator according to the procedure described in (see 9.2.6.7.1).
- c) This modulation symbol shall be modulated with power density P_{ODCCH} on subcarrier f in Orthogonal Frequency Division Multiplexing symbol t within tile q , i.e., the value of the corresponding subcarrier shall be $\sqrt{P_{\text{ODCCH}}} s$, where P_{ODCCH} is the power density assigned the Reverse OFDMA Dedicated Control Channel by the RCC MAC Protocol (8.7).
- d) Increment f by 1. If $f = N_{\text{BLOCK}}/2$, set $f = 0$ and increment t by 1. If $t = N_{\text{FRAME}}/2$, set $t = 0$ and increment q by 1.
- e) Repeat steps b)–d) until $q = 2$.

Note that in the above procedure, the Reverse OFDMA Dedicated Control Channel modulation symbols are placed in frequency first, then followed by time.

9.3.1.3.4.3 Reverse Acknowledgment Channel

The Reverse Acknowledgment Channel (R-ACKCH) is used to acknowledge Forward Link PHY frames transmitted on the Forward Data Channel. For the purpose of this subclause, the sector of interest is the Forward Link Serving Sector (FLSS), which may or may not be the same as the Reverse Link Serving Sector (RLSS). For convenience of notation, the phrase “of the Forward Link Serving Sector” shall be omitted. Sector-dependent quantities such as PilotPN, hop-permutations etc., used in this subclause shall be interpreted as “PilotPN of the FLSS,” “hop-permutations of the FLSS,” etc.

9.3.1.3.4.3.1 Reverse Acknowledgment Channel Resource assignment

Reverse Acknowledgment Channel transmissions are determined by a RACKNodeIndex and a corresponding RACKVal specified by the RCC MAC Protocol (see 8.7). An access terminal may be assigned zero, one, or more RACKNodeIndex values in any Reverse Link PHY frame.

An access terminal that is assigned an RACKNodeIndex DRACKCH shall be assigned four Reverse Acknowledgment Channel resources according to the following procedure:

- a) Set $g = \lfloor D_{\text{RACKCH}} / N_{\text{PARTIAL-TILES}} \rfloor$ and $u = D_{\text{RACKCH}} \bmod N_{\text{PARTIAL-TILES}}$.
- b) Set $\text{SEED}_{\text{RACKCH-ROWS}} = f_{\text{PHY-HASH}}(3 \times 64 + (g \bmod 64))$.
- c) Generate a permutation $H^{\text{ij}}_{\text{RACKCH-ROWS}}$ of size N_{TILES} using the common permutation generation algorithm described in 9.2.5.1 with seed $\text{SEED}_{\text{RACKCH-ROWS}}$.
- d) Set $\text{SEED}_{\text{RACKCH-COLS}} = f_{\text{PHY-HASH}}(2 \times 64 + (g \bmod 64))$.
- e) Generate a permutation $H^{\text{ij}}_{\text{RACKCH-COLS}}$ of size N_{SUBTILES} using the common permutation generation algorithm described in 9.2.5.1 with seed $\text{SEED}_{\text{RACKCH-COLS}}$.
- f) Set $\text{SEED}_{\text{RACKCH-CODES}} = f_{\text{PHY-HASH}}(64)$.
- g) Generate a permutation $H^{\text{ij}}_{\text{RACKCH-CODES}}$ of size $L/2$ using the common permutation generation algorithm described in 9.2.5.1 with seed $\text{SEED}_{\text{RACKCH-CODES}}$.
- h) Initialize a counter k to 0. Repeat the following steps until $k = N_{\text{SUBTILES}}$.
 - 1) Compute $t = (u - k) \bmod N_{\text{PARTIAL-TILES}}$
 - 2) Set $t_{\text{TILE}} = H^{\text{ij}}_{\text{RACKCH-ROWS}}(t)$, $k_{\text{SUBTILE}} = H^{\text{ij}}_{\text{RACKCH-COLS}}(k \bmod N_{\text{SUBTILES}})$ and $w = 2^{\text{H}^{\text{ij}}_{\text{RACKCH-CODES}}((g + t_{\text{TILE}}N_{\text{SUBTILES}} + k_{\text{SUBTILE}}) \bmod (8))}$.
 - 3) Assign the Reverse Acknowledgment Channel resource $(t_{\text{TILE}}, k_{\text{SUBTILE}}, w)$ to the access terminal.
 - 4) Increment k by 1.

An example of the resource assignment of the Reverse Acknowledgment Channel modulation and subtile mapping is shown in Figure 111.

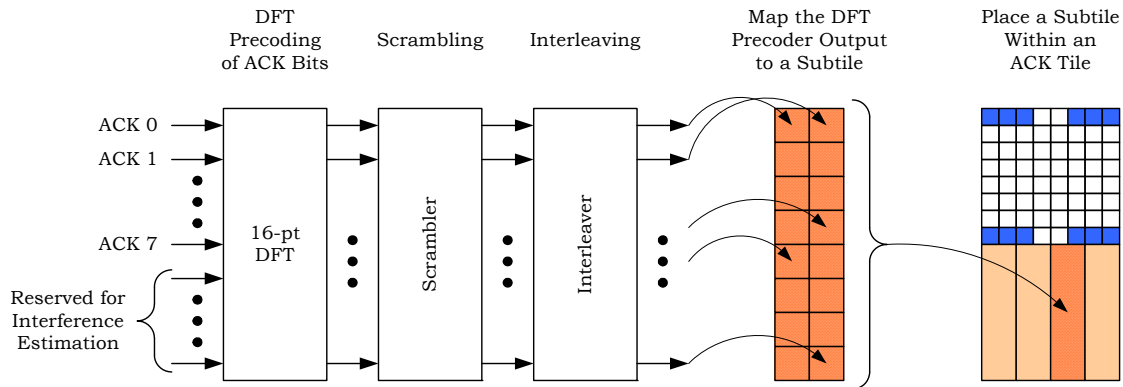


Figure 111—Reverse Acknowledgment Channel modulation and subtile mapping

9.3.1.3.4.3.2 Reverse Acknowledgment Channel Resource assignment example

The access terminal is assigned four subtiles to transmit its Reverse Acknowledgment Channel. The notion of subtiles is defined in 9.2.11.3. The algorithm in the previous subclause ensures that every subtile either gets assigned to only one access terminal or, if it is assigned to more than one access terminal, the access terminals use different exponential sequences.

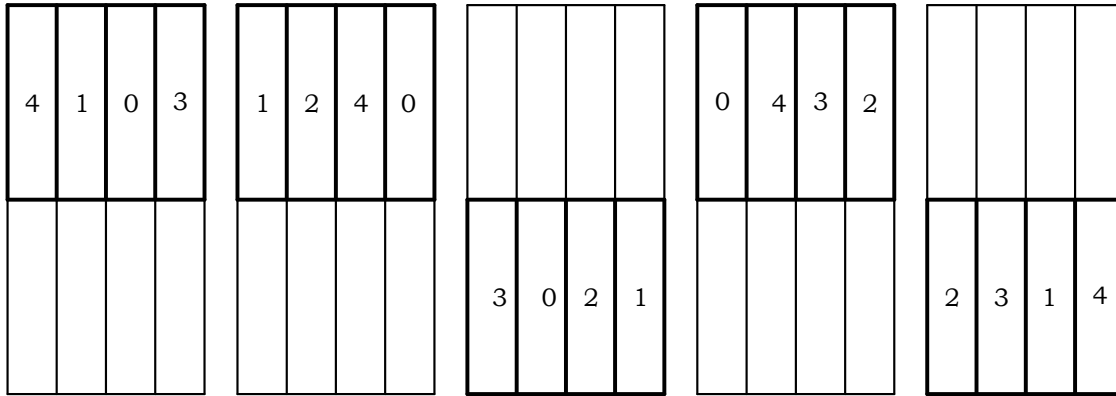


Figure 112—Reverse Acknowledgment Channel Resource assignment

Figure 112 shows an example of the Reverse Acknowledgment Channel Resource assignments to five access terminals. There are five partial-tiles, each divided into four subtiles as shown in Figure 112. Assume that $H_{\text{RACKCH-ROWS}}^i: \{0, 1, 2, 3, 4\} \rightarrow \{1, 2, 0, 4, 3\}$ and $H_{\text{RACKCH-COLS}}^i: \{0, 1, 2, 3\} \rightarrow \{2, 1, 3, 0\}$. Assume that the DRACKCH values of the five access terminals are such that the access terminals correspond to $g = 0$ and their u values are 2, 3, 1, 4, and 0 respectively. The assignments of subtiles are depicted in Figure 112. For instance, the access terminal 0 has $g = 0$ and $u = 2$, and is assigned to the partial-tile values of $\{0, 2, 1, 3\}$ and the corresponding subtile values of $\{2, 1, 3, 0\}$. If there is a sixth access terminal with $D_{\text{RACKCH}} = 5$, it shall have the same value of u as one of the first five access terminals and shall therefore share the four subtiles with that access terminal. However, for that access terminal, the value of g shall be different, and therefore the access terminal shall use a different value of ω and hence a different exponential sequence.

9.3.1.3.4.3.3 Reverse Acknowledgment Channel modulation

An access terminal shall transmit a sequence $X_{ACK}(t_{TILE}, k_{SUBTILE}, \omega)$ on each Reverse Acknowledgment Channel resource $(t_{TILE}, k_{SUBTILE}, \omega)$ assigned to it. The sequence $X_{ACK}(t_{TILE}, k_{SUBTILE}, \omega)$ is an ON-OFF transmission specified by a bit RACKVal defined by the RCC MAC Protocol (see 8.7) for each RACKBaseNodeIndex assigned to the access terminal.

When RACKVal is equal to 1, the sequence $X_{\text{ACK}}(t_{\text{TILE}}, k_{\text{SUBTILE}}, \omega)$ shall be

$$X_{ACK}(t_{TILE}, k_{SUBTILE}, \omega) = \sqrt{P_{RACKCH}} S_{ACK}^L E_{\omega}^L$$

where E_{ω}^L is the exponential sequence of length L as defined in 9.2.11.3. and P_{RACKCH} is the power density allocated to the Reverse Acknowledgment Channel by the RCC MAC Protocol (see 8.7). S_{ACK}^L is the sequence of length L generated using the common complex scrambling algorithm described in 9.2.5.2 with input seed equal to $f_{\text{PHY-HASH}}(5 \times 2^{20} + \text{SectorSeed})$, where the SectorSeed corresponds to the sector of interest and is defined in 9.2.2.2.3. The expression $S_{\text{ACK}}^L E_{\omega}^L$ is used to denote the point-wise multiplication of the sequences S_{ACK}^L and E_{ω}^L .

When RACKVal is equal to 0, the sequence $X_{ACK}(t_{TILE}, k_{SUBTILE}, \omega)$ shall be a sequence of L zeros.

The sequence $X_{ACK}(t_{TILE}, k_{SUBTILE}, \omega)$ shall be used to modulate the L subcarriers in the subtitle (t_{TILE} , $k_{SUBTILE}$) according to the following procedure:

- a) Initialize an Orthogonal Frequency Division Multiplexing symbol counter t to t_{START} , where t_{START} is the lowest indexed Orthogonal Frequency Division Multiplexing Symbol in the subtile. Initialize a subcarrier counter f to f_{START} , where f_{START} is the lowest indexed subcarrier in the subtile. Initialize a modulation symbol counter i to 0.
- b) Repeat the following steps until $i = L$.
 - 1) Let $j = H_{\text{ACKCH-INTERLEAVE}}(i)$, where $H_{\text{ACKCH-INTERLEAVE}}$ is a permutation of size L such that $H_{\text{ACKCH-INTERLEAVE}}(x) = 8\lfloor x/8 \rfloor + \text{BR}_3(x \bmod 8)$. BR_3 denotes the bit-reversed value of x when x is expressed as a 3-bit quantity.
 - 2) Modulate the subcarrier f in Orthogonal Frequency Division Multiplexing Symbol t with modulation symbol $X_{\text{ACK}}^j(t_{\text{TILE}}, k_{\text{SUBTILE}}, \omega)$, where $X_{\text{ACK}}^j(t_{\text{TILE}}, k_{\text{SUBTILE}}, \omega)$ is the j^{th} element in the sequence $X_{\text{ACK}}(t_{\text{TILE}}, k_{\text{SUBTILE}}, \omega)$.
 - 3) Increment t by 1. If $t = t_{\text{START}} + 1$, set t to t_{START} and increment f by 1.
 - 4) Increment i by 1.

9.3.1.3.4.4 Reverse Orthogonal Frequency Division Multiple Access Data Channel

The Reverse Orthogonal Frequency Division Multiple Access Data Channel (R-ODCH) consists of either a data packet or an erasure sequence, both of which can span one or more Reverse Link PHY frames. The set of Reverse Link PHY frames on which this packet or erasure sequence is transmitted is determined by Clause 8. Each data packet and erasure sequence is also assigned a set of hop-ports in each PHY frame of transmission by Clause 8. Each data packet is further associated with a packet format index, which is also assigned by the RTC MAC Protocol (see 8.6).

9.3.1.3.4.4.1 Reverse Orthogonal Frequency Division Multiple Access Data Channel Data packet encoding

Each Reverse Orthogonal Frequency Division Multiple Access Data Channel packet is generated by the RTC MAC Protocol (see 8.6), and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled, and modulated according to the procedure described in 9.2.6.1. A CRC length of $N_{\text{CRC,DATA}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(7 \times 2048 \times \text{SectorSeed} + m \bmod 2048)$ shall be used for the data scrambling operation, where the SectorSeed corresponds to the RLSS and is defined in 9.2.2.2.3, m is the MACID of the access terminal as assigned by the RLSS. In the case of Reverse Link PHY frame 0 in the superframe when DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 0, the scrambling register shall be re-initialized with the above seed after the first eight modulation symbols have been modulated.

9.3.1.3.4.4.2 Reverse Orthogonal Frequency Division Multiple Access Data Channel Data packet modulation

The data packet shall be modulated on to the hop-ports assigned to this packet according to the following procedure:

- a) Initialize a port counter i , a HARQ retransmission counter r , a frame counter f , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- b) Let $F(r)$ be the total number of PHY frames to be used by the r^{th} HARQ retransmission of the packet, as specified by Clause 8. The frames shall be indexed $(r, 0), (r, 1) \dots (r, F(r) - 1)$.
- c) Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) . Let the resulting sequence be denoted by $p_0, p_1, \dots, p_{n-1}$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .
- d) Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} Orthogonal Frequency Division Multiplexing symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY frame (r, f) , which is a function of the packet format and HARQ transmission index r . If n_{sc} is not

assigned to the Reverse Acknowledgment Channel of the RLSS or the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel of the RLSS, is not a DPICH subcarrier and is not part of the RL Silence Interval, then a modulation symbol s from subpacket m with modulation order q is generated by the modulator according to the procedure described in (Clause 8), where

$m = (i_{\text{TILE}} + (j \bmod N_{\text{FRAME}} + i \bmod N_{\text{BLOCK}}) \bmod N_{\text{SUBPACKETS-IN-TILE}}) \bmod t$, t is the total number of subpackets in the packet, N_{BLOCK} is the number of subcarriers in a block, $i_{\text{TILE}} = \lfloor i/N_{\text{BLOCK}} \rfloor$ and $N_{\text{SUBPACKETS-IN-TILE}}$ is computed as follows:

$$N_{\text{SUBPACKETS-IN-TILE}} = t \text{ if } i_{\text{TILE}} < (N_{\text{TILES}} \bmod t), \text{ where } N_{\text{TILES}} = \lfloor n/N_{\text{BLOCK}} \rfloor$$

$$N_{\text{SUBPACKETS-IN-TILE}} = \min \left(t, \left\lceil \frac{16t}{N_{\text{TILES}} - (N_{\text{TILES}} \bmod t)} \right\rceil \right)$$

- e) This modulation symbol shall be modulated with energy P_{ODCH} on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{P_{\text{ODCH}}} s$, where P_{ODCH} is the power specified for the tile with index i_{TILE} in PHY frame (r, f) and is determined by the RTC MAC Protocol.
- f) Increment i . If $i = n$, increment j and set $i = 0$.
- g) Increment f and set $j = 0$ if any of the following two conditions is satisfied:
 - 1) If this is a Reverse Link PHY frame with index 0 within the superframe and if $j = N_{\text{FRAME}} + N_{\text{PREAMBLE}}$.
 - 2) For any other Reverse Link PHY frame, if $j = N_{\text{FRAME}}$.
- h) If $f = F(r)$, then increment r and set $f = 0$.
- i) If the last HARQ retransmission has been completed (as determined by the RTC MAC Protocol), then stop. Else repeat steps b) through h).

9.3.1.3.4.4.3 Reverse Orthogonal Frequency Division Multiple Access Data Channel erasure sequence transmission

When the RTC MAC Protocol (see 8.8) instructs the Physical Layer to transmit an erasure sequence in a given Reverse Link PHY frame, the erasure sequence shall be modulated according to the following procedure:

- a) Construct a one-bit packet, with the bit in the packet being set to zero. This packet is encoded, channel interleaved, repeated, scrambled, and modulated according to the procedure described in 9.3.1.3.4.4.1.²⁸ A seed equal to $f_{\text{PHY-HASH}}(2048 \times p + m)$ shall be used for the data scrambling operation, where p is the SectorSeed corresponding to the RLSS in the Reverse Link PHY frame of interest, and m is the MACID of the access terminal in the RLSS. QPSK modulation shall be used for all of the modulation symbols in the packet.
- b) Let $N_{\text{ER}} = \min(N_{\text{TOTAL-HOP-PORTS}}, N_{\text{ERASURESEQ,MAX}})$, where $N_{\text{ERASURESEQ,MAX}}$ is equal to 32 and $N_{\text{TOTAL-HOP-PORTS}}$ is the total number of usable hop-ports assigned to access terminal as specified by the RTC MAC Protocol (see 8.8). Arrange the $N_{\text{TOTAL-HOP-PORTS}}$ hop-ports in sequential order. Let the resulting sequence be denoted by $q_0, q_1, q_2 \dots$.
- c) Define an erasure-available hop-port as a hop-port not mapped to the Reverse Acknowledgement Channel and the Reverse OFDMA Dedicated Control Channel. Let $N_{\text{TOTAL-USABLE-HOP-PORTS}}$ be

²⁸The operations before scrambling and modulation are all trivial operations, i.e., they result in an all-zeros sequence. The erasure sequence is equivalent to scrambling an all-zeros sequence of the required length, followed by QPSK modulation.

the number of erasure-available hop-ports. If $N_{\text{TOTAL-USABLE-HOP-PORTS}} \leq N_{\text{ER}}$, set $n = N_{\text{TOTAL-USABLE-HOP-PORTS}}$. If $N_{\text{TOTAL-USABLE-HOP-PORTS}} > N_{\text{ER}}$, then let $N_{\text{MIN-ERASURE-BLOCKS}}$ be the minimum number of hop-port blocks such that the set of erasure available hop-ports in the lowest indexed $N_{\text{MIN-ERASURE-BLOCKS}}$ is at least N_{ER} . Set n to be the number of erasure-available hop-ports in the $N_{\text{MIN-ERASURE-BLOCKS}}$ hop-port blocks. Let $p_0, p_1, \dots, p_{n-1}$ be the n erasure-available hop-ports.

- d) Initialize the port counter i to 0 and initialize an OFDM symbol counter j to 0.
- e) Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} OFDM symbol in PHY frame (r, f) . If n_{sc} is not assigned to the Reverse Acknowledgment Channel of the RLSS or the Reverse OFDMA Dedicated Control Channel of the RLSS and is not a Reverse Dedicated Pilot Channel subcarrier, then a QPSK modulation symbol s is generated by the modulator according to the procedure described in 9.2.6.7.1.
- f) This modulation symbol shall be modulated with energy P_{ODCH} on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{N_{\text{ER}}/P_{\text{ODCH}}} s$. Here, P_{ODCH} is the power specified for the erasure sequence in the Reverse Link in PHY frame and is determined by the RTC MAC Protocol (see 8.7).
- g) Increment i . If $i = n$, increment j and set $i = 0$.
- h) Halt this procedure if either of the following two conditions is satisfied:
 - 1) If this is a Reverse Link PHY frame with index 0 within the superframe and if $j = N_{\text{FRAME}} + N_{\text{PREAMBLE}}$.
 - 2) If $j = N_{\text{FRAME}}$ for any other Reverse Link PHY frame.
- i) Repeat steps e) through h).

9.3.1.4 Synchronization and timing

The access terminal shall follow the synchronization requirements defined in 9.2.2.1.

9.3.1.4.1 Transmit burst timing synchronization requirements

At the AT, in Wideband TDD mode, no burst timing requirement is necessary because the AT is synchronized with the AN (see 9.3.1.5.1). To prevent accumulation of long term errors, the AT should adaptively correct its timing based on the observed AN waveform.

Burst timing synchronization does not apply to FDD mode.

9.3.1.5 Transmitter performance

9.3.1.5.1 Frequency control requirements

At the access terminal, both the transmitted center frequency and the symbol clock frequency shall be synchronized to the AN with a tolerance not to exceed $\pm 2\%$ of the subcarrier spacing.

9.3.1.5.2 AT transmit waveform quality

The goal is to make sure that the receiver at AN does not degrade by more than 0.5 dB in performance due to transmitter imperfections. The metric used is denoted by mean square constellation error (MSCE), which is basically the mean square error between the ideal constellation point and the actual one to be received after considering transmit imperfections.

9.3.1.5.2.1 Procedure for AT MSCE measurement

The following procedure shall be followed:

- a) The AT is assigned all tiles for data except for those designated for guard band
- b) The DC tone shall not be populated
- c) The transmitted signal is cable-connected to the receiver with two receive antennas. Denote the received samples on antennas 1 and 2 by y_1 and y_2 respectively
- d) After down conversion, the receiver performs time and frequency offsets calculation
- e) Time and frequency correction shall be applied to the received signal yielding to samples z_1 and z_2 from antenna 1 and antenna 2 respectively
- f) Using pilots, for each data tone, estimate the average channel per Rx antenna, denote the estimates by h_1 and h_2 for antenna 1 and antenna 2 respectively
- g) The actual received constellation point on that tone is given by $\hat{x} = \frac{(h_1^* z_1 + h_2^* z_2)}{(|h_1|^2 + |h_2|^2)}$

where a^* is the complex conjugate of a and $|a|$ is the absolute value of a

- h) Find the closest constellation point to $\hat{x}$ in Euclidean distance and denote that point by x
- i) Find the MSCE metric given by

$$\varepsilon = \frac{\sum_{j=1}^{N_f} \sum_{k=1}^{N_p} (x_I(j, k) - \hat{x}_I(j, k))^2 + (x_Q(j, k) - \hat{x}_Q(j, k))^2}{\sum_{j=1}^{N_f} \sum_{k=1}^{N_p} (x_I(j, k))^2 + (x_Q(j, k))^2}$$

where a_I, a_Q are the real and imaginary parts of a , N_p is the number of modulation symbols in the packet, and N_f is the number of frames used for averaging MSCE.

The MSCE shall not exceed the values in Table 213.

Table 213—MSCE values as a function of packet format

Packet Format (PF) – Transmission (see Table 170)	Modulation	MSCE (dB)
2 – 1 st transmission	QPSK	–15
3 – 1 st transmission	8-PSK	–18
4 – 1 st transmission	16-QAM	–21
6 – 1 st transmission	64-QAM	–26

9.3.2 Receiver**9.3.2.1 Channel spacing and designation**

Channel spacing and designation for the access terminal reception shall be as specified in 9.3.1.1.1. Valid channels for operations shall be as specified in 9.3.1.1.1.

9.3.2.2 Demodulation characteristics

9.3.2.2.1 Processing

The access terminal demodulation process shall perform complementary operations to the access network modulation process on the Forward Channel (see 9.2.6).

9.3.2.3 Limitations on emissions

The system shall meet all applicable regulatory and legal requirements.

9.3.2.4 Receiver performance requirements

9.3.2.4.1 Receiver sensitivity requirements

The low sensitivity level is defined for PF 2 first transmission (which corresponds to a spectral efficiency of 0.94 and QPSK modulation), a PER of 1%, pilot format 1, and utilizing all tones except for those designated for guard band. The low sensitivity level shall not exceed the values defined in Table 214.

Table 214—Low sensitivity level

Bandwidth (MHz)	Sensitivity (dBm)
5	−94
10	−91
20	−88

The high sensitivity level is defined for PF 6 first transmission (which corresponds to a spectral efficiency of 3.0 and 64-QAM modulation), a PER of 1%, pilot format 1, and utilizing all tones except for those designated for guard band. The high sensitivity level shall not exceed the values defined in Table 215.

Table 215—High sensitivity level

Bandwidth (MHz)	Sensitivity (dBm)
5	−83
10	−80
20	−77

The received power level is averaged over data frames only and hence the above definition works for FDD and TDD.

9.3.3 Malfunction detection

9.3.3.1 Malfunction timer

The access terminal shall have a malfunction timer that is separate from and independent of all other functions and that runs continuously whenever power is applied to the transmitter of the access terminal. Sufficient reset commands shall be interspersed throughout the access terminal logic program to ensure that the timer never expires as long as the proper sequence of operations is taking place. If the timer expires, a malfunction shall be assumed and the access terminal should be inhibited from transmitting. The maximum time allowed for expiration of the timer is two seconds.

9.3.3.2 False Transmission

A protection circuit shall be provided to minimize the possibility of false transmitter operation caused by component failure within the access terminal.

9.4 Access network requirements

This subclause defines requirements specific to access network equipment and operation.

9.4.1 Transmitter

The transmitter shall reside in each sector of the access network. These requirements apply to the transmitter in each sector.

Each sector is assigned an integer identifier in the range 0–511 (including 0 and 511) called the PilotPN.

9.4.1.1 Frequency parameters

9.4.1.1.1 Channel spacing and designation

See 3GPP2 C.S0057-B for a description of the band classes that an access network may support.

9.4.1.2 Modulation characteristics

9.4.1.2.1 Forward channel signals

The Forward Link channels are described in Table 216.

Table 216—Description of the Forward Link Channels

Pilot Channels	
F-CPICH	Forward Common Pilot Channel
F-CQIPICH	Forward Channel Quality Indicator Pilot Channel
F-SCQIPICH	Forward Superposed Channel Quality Indicator Pilot Channel
F-DPICH	Forward Dedicated Pilot Channel
F-PPICH	Forward Preamble Pilot Channel
F-BPICH	Forward Beacon Pilot Channel
F-CNCH	Forward Cell Null Channel

Table 216—Description of the Forward Link Channels (continued)

Control Channels Transmitted in the Superframe Preamble	
F-ACQCH	Forward Acquisition Channel
F-PBCCH	Forward Primary Broadcast Control Channel: Carries Deployment-Specific Parameters
F-SBCCH	Forward Secondary Broadcast Control Channel: Carries Sector-Specific Parameters
F-QPCH	Forward Quick Paging Channel
F-OSICH	Forward Other-Sector-Interference Channel: Carries an Other-Sector-Interference Indication

Control Channels Transmitted in the Control Segment of PHY frames	
F-SCCH	Forward Shared Control Channel: Carries Access Grants, Assignment Messages, and Other Messages Related to Resource Management
F-ACKCH	Forward Acknowledgement Channel: Carries Acknowledgement Bits for the Reverse Link HARQ Transmissions
F-PCCH	Forward Power Control Channel: Carries Reverse Link Power Control Commands
F-PQICH	Forward Pilot Quality Indicator Channel: Carries the Strength of the Reverse Link Pilots of Each Active Terminal
F-FOSICH	Forward Fast Other-Sector-Interference Channel: Carries an Other-Sector-Interference Indication Transmitted at a Faster Rate But with Less Coverage Than the Forward Other Sector Interference Channel
F-SPCH	Forward Start of Packet Channel
F-RABCH	Forward Reverse Activity Bit Channel
F-IOTCH	Forward Interference over Thermal Channel

Traffic Channel	
F-DCH	Forward Data Channel: Carries the Forward Link Data
F-BMSCH	Forward Broadcast and Multicast Services Channel
F-SDCH	Forward Superposed Data Channel

9.4.1.2.1.1 Channel structures

The structure of the Forward Primary Broadcast Control Channel is shown in Figure 113.

The structure of the Forward Secondary Broadcast Control Channel is shown in Figure 114. The structure of the Forward Quick Paging Channel is shown in Figure 115. The structure of the Forward Acknowledgment Channel is shown in Figure 116. The structure of the Forward Start-of-Packet Channel is shown in Figure 117. The structure of the Forward Shared Control Channel is shown in Figure 118. The structure of the Forward Pilot Quality Indicator Channel is shown in Figure 119. The structure of the Forward Reverse Activity Bit Channel is shown in Figure 120. The structure of the Forward Fast Other Sector Interference Channel is shown in Figure 121. The structure of the Forward Interference over Thermal Channel is shown in Figure 122. The structure of the Forward Data Channel is shown in Figure 123.

The channel structure for the superframe preamble is as shown in Figure 124. The channel structure for the Physical Layer frames is as shown in Figure 125. The transmit chain for the Single Input Single Output case

is shown in Figure 126. Space Time Transmit Diversity for 2 antennas is shown in Figure 127. Space Time Transmit Diversity for 4 antennas in Figure 128. The generic Multiple Input Multiple Output transmitter is in Figure 129. The layer permutation is described in Figure 130. The precoding operation is shown in Figure 131. The SDMA operation is shown in Figure 132.

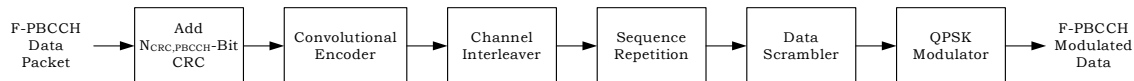


Figure 113—Channel structure for Forward Primary Broadcast Channel

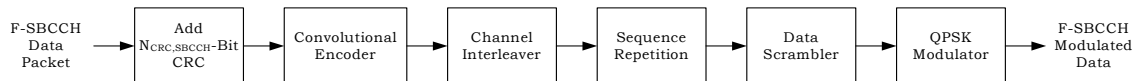


Figure 114—Channel structure for F-SBCCH

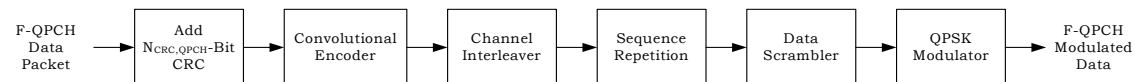


Figure 115—Channel structure for Forward Quick Paging Channel

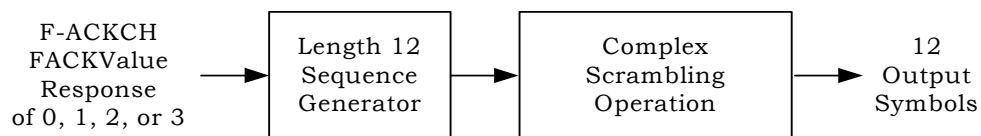


Figure 116—Channel structure for Forward Acknowledgement Channel

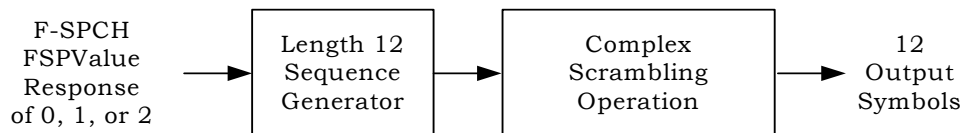


Figure 117—Channel structure for Forward Start-of-Packet Channel

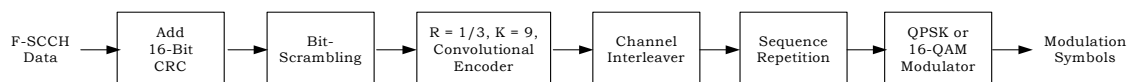


Figure 118—Channel structure for Forward Shared Control Channel

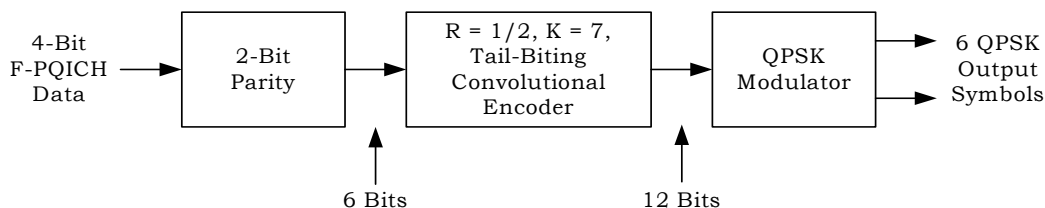


Figure 119—Channel structure for Forward Pilot Quality Indicator Channel



Figure 120—Channel structure for Forward Reverse Activity Bit Channel

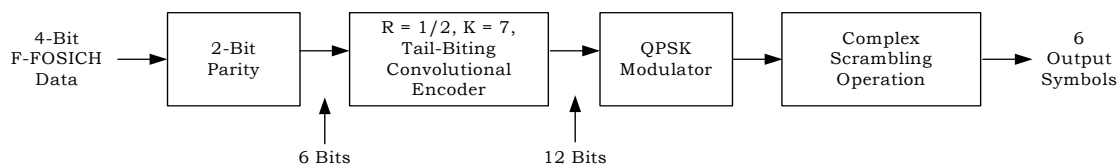


Figure 121—Channel structure for Forward Fast Other Sector Interference Channel

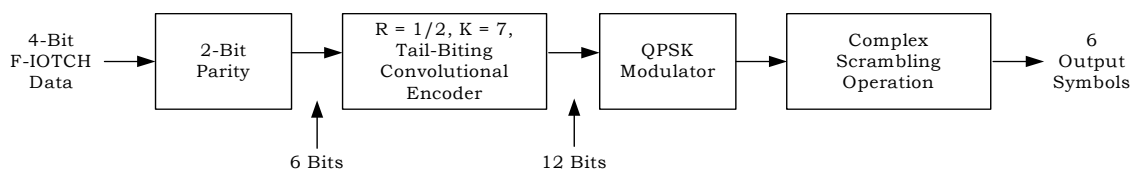


Figure 122—Channel structure for Forward Interference over Thermal Channel

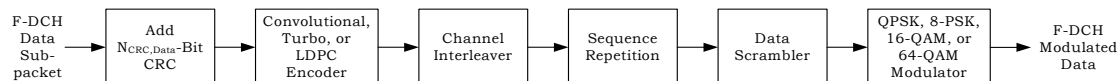
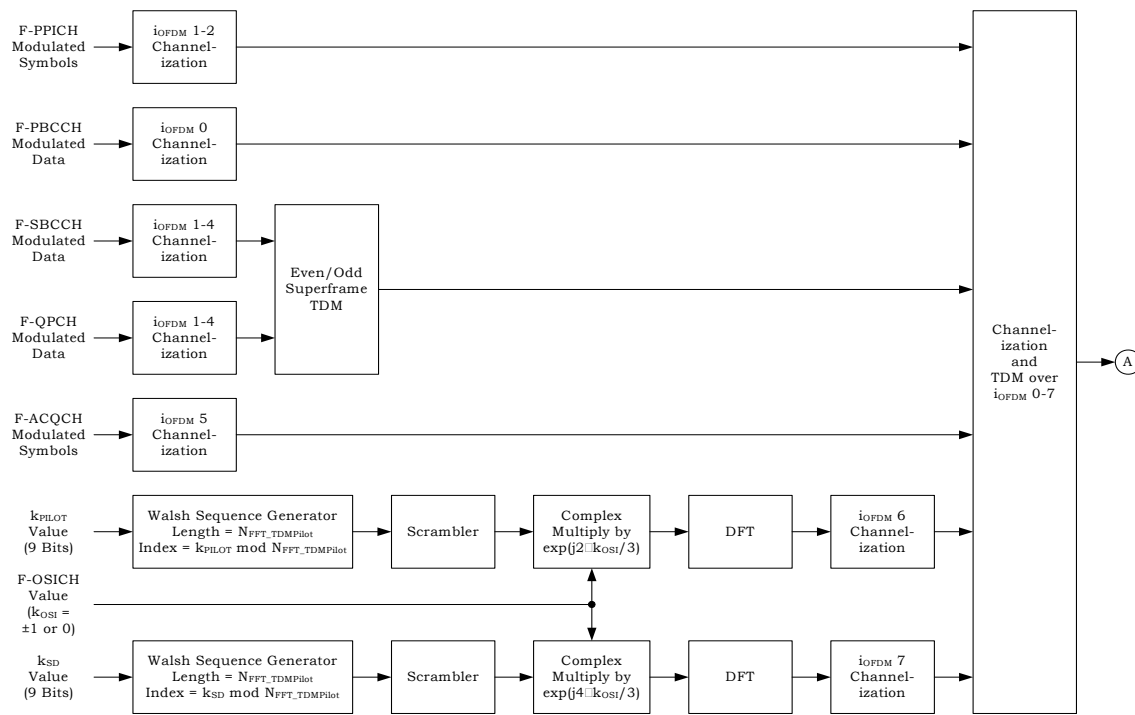


Figure 123—Channel structure for Forward Data Channel



Note: ioFDM is the superframe OFDM symbol index.

Figure 124—Channel structure in the superframe preamble

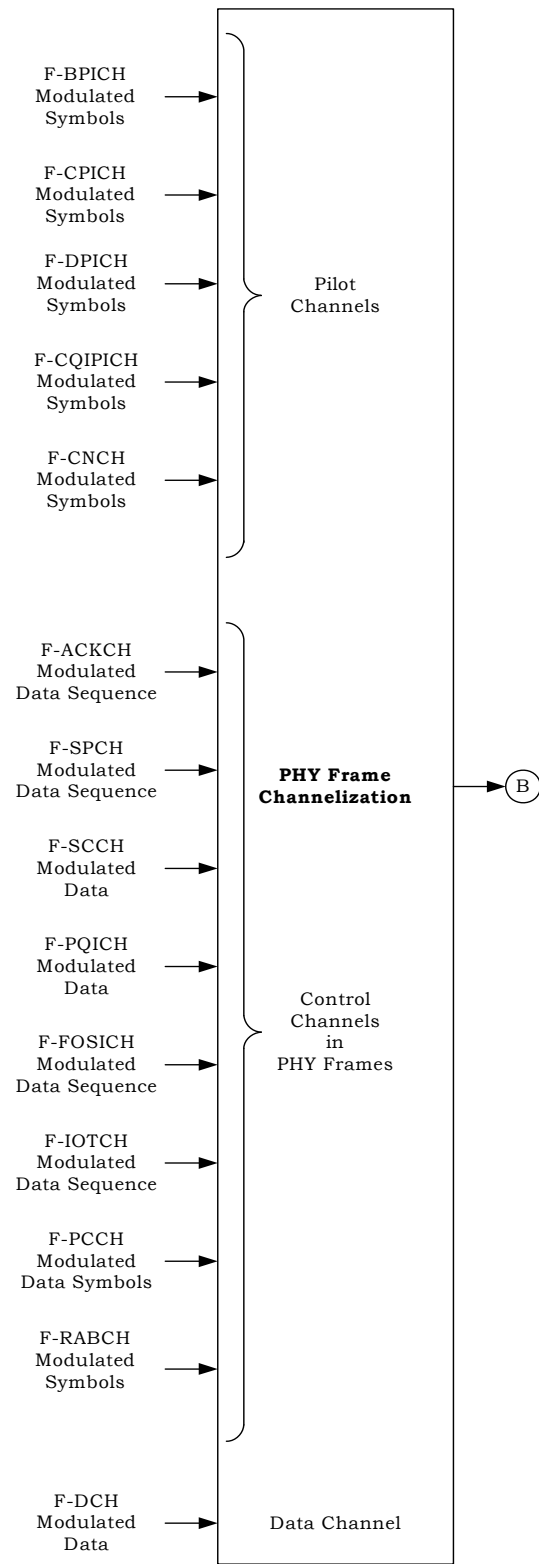


Figure 125—Channel structure of the PHY frames

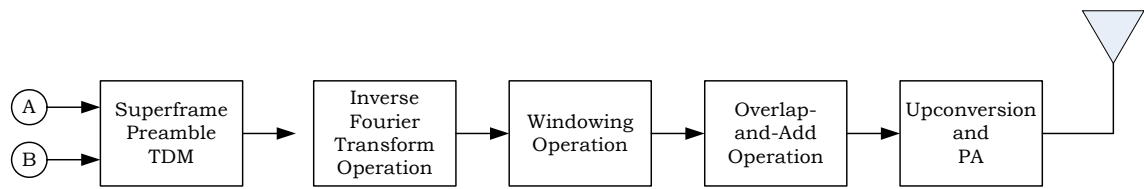


Figure 126—Channel structure for the single-transmit-antenna case

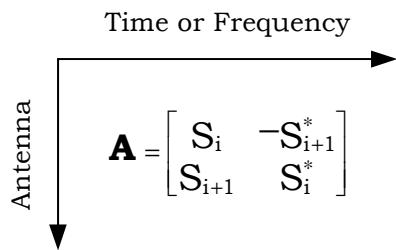


Figure 127—Space time transmit diversity—two effective antennas

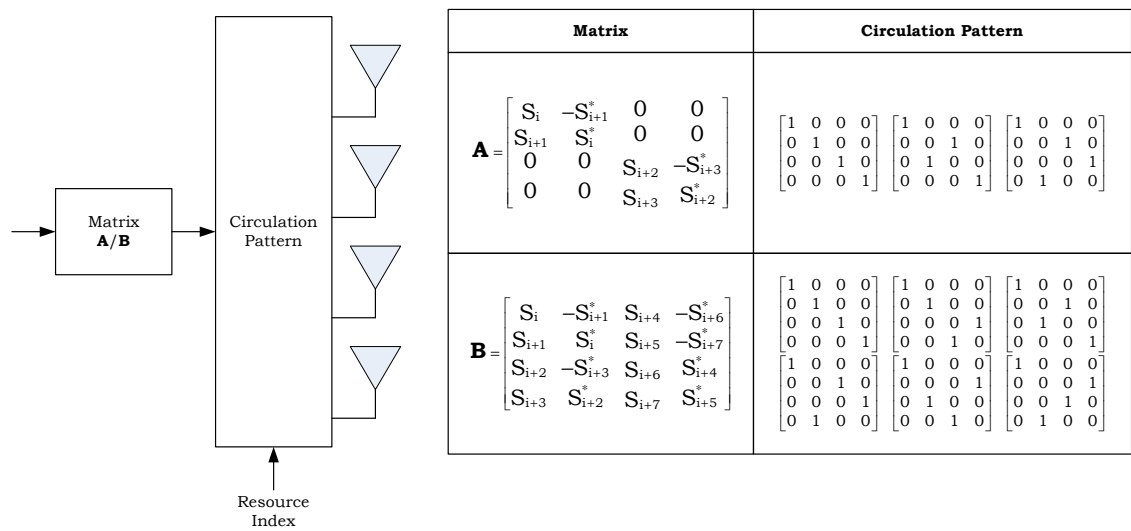


Figure 128—Space time transmit diversity—four effective antennas

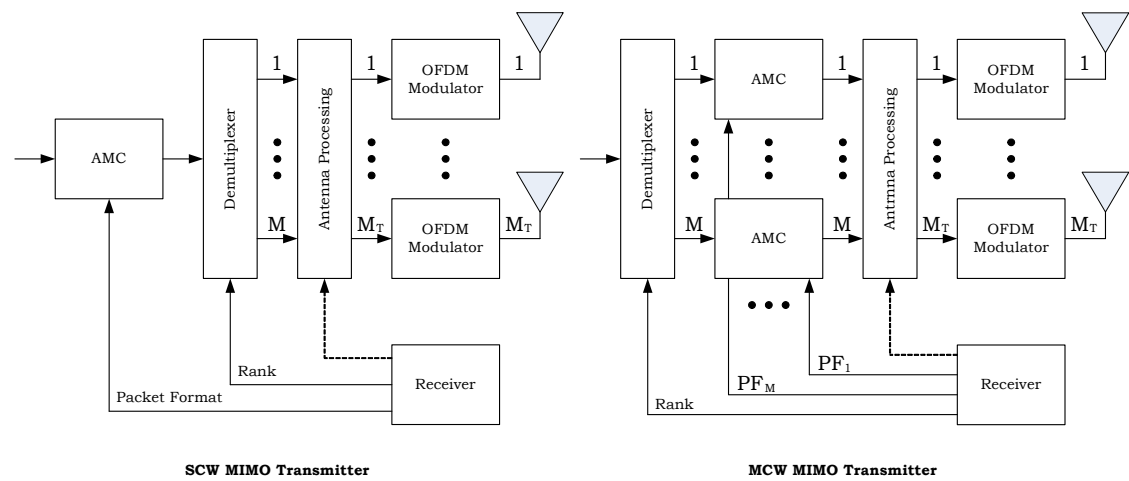


Figure 129—Generic multiple input multiple output transmitter

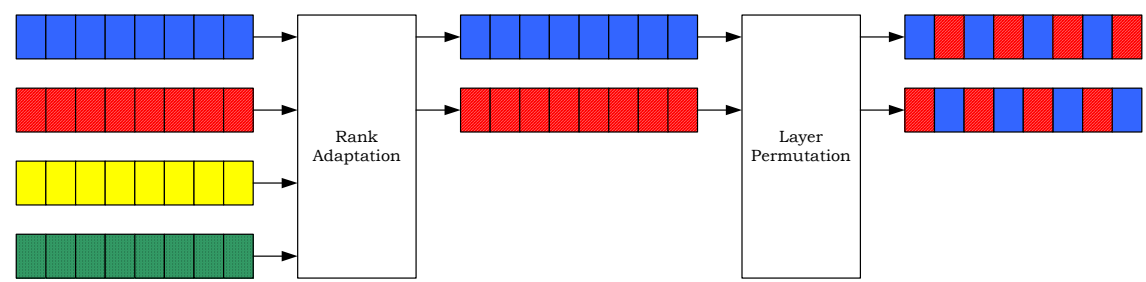


Figure 130—Layer permutation for multi-codeword multiple input multiple output

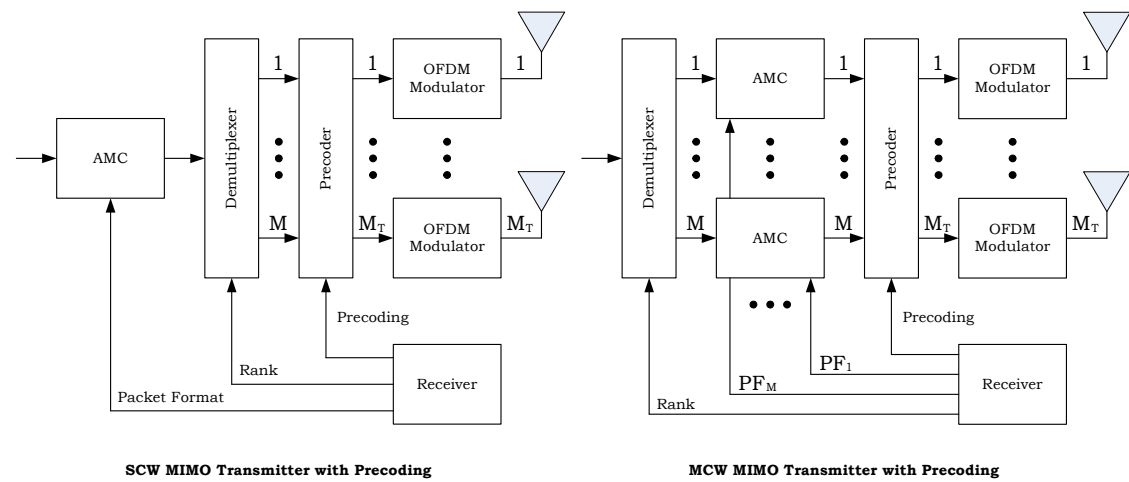


Figure 131—Precoding for Forward Data Channel

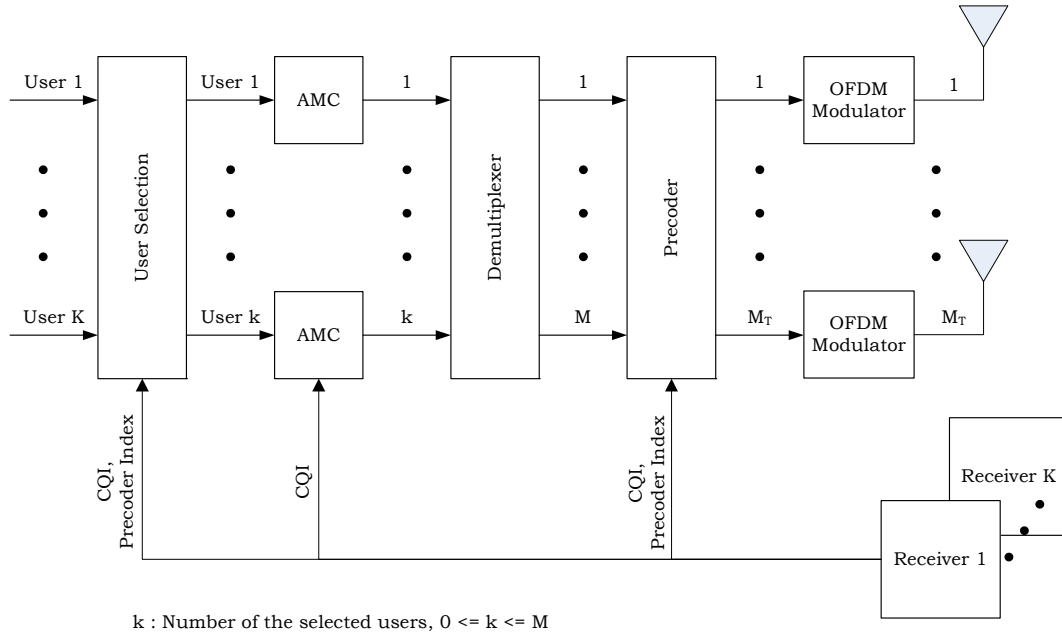


Figure 132—SDMA for Forward Data Channel

9.4.1.2.2 Channels in the Superframe Preamble

Channels in the superframe preamble shall be transmitted on a fraction of the bandwidth. For this purpose, a quantity $N_{\text{FFT,TDMPilot}}$ is defined as $N_{\text{FFT,TDMPilot}} = 512$. Furthermore,

$$P_{\text{PREAMBLE}} = \frac{N_{\text{FFT}}}{\min(N_{\text{FFT}} - N_{\text{GUARD}}, N_{\text{FFT,TDMPilot}})} \text{ and}$$

$$P_{\text{PREAMBLE,TDMPilot}} = \frac{N_{\text{FFT}}}{\min((N_{\text{FFT}} - N_{\text{GUARD}})/4, N_p)}$$

are defined, where N_p is the number of pilots transmitted in the Orthogonal Frequency Division Multiplexing symbol with index 5 in the superframe preamble and is defined in Table 217. Define the amplitude of the signal $A_{\text{PREAMBLE}} = \sqrt{P_{\text{PREAMBLE}}}$. Note that all occupied subcarriers in the first five OFDM symbols of the superframe preamble (namely the symbols carrying the Forward Primary Broadcast Control Channel, the Forward Secondary Broadcast Control Channel, the Forward Quick Paging Channel, and the Forward Preamble Pilot Channel) are transmitted at unit power i.e., the transmit power of these channels acts as a reference for all the other channels.

9.4.1.2.2.1 Forward Acquisition Channel

TDM Pilot 1 forms the Forward Acquisition Channel, and is transmitted on the Orthogonal Frequency Division Multiplexing symbol with index 5 in the superframe preamble. For FFT sizes of 1024 and 2048, TDM Pilot 1 spans only the central 480 subcarriers of this Orthogonal Frequency Division Multiplexing symbol, and occupies every fourth subcarrier over this span.

More precisely, define $sc\_start = \max(N_{\text{GUARD,LEFT}}, 16, N_{\text{FFT}}/2 - 240)$, $sc\_end = \min(sc\_start + 4 \times N_p, N_{\text{FFT}} - N_{\text{GUARD,RIGHT}}, N_{\text{FFT}}/2 + 240)$, and $sc\_offset = 16 + \max(0, N_{\text{FFT}}/2 - 256)$. The values of the

complex modulation symbols, X_i , $i = 0$ to $N_{\text{FFT}} - 1$, for the TDM Pilot 1 Orthogonal Frequency Division Multiplexing symbol shall be given by

$$X_i = \begin{cases} G_{\text{LOWPAR-GAIN}} \sqrt{P_{\text{PREAMBLE, TDM Pilot 1}}} \exp\left(-j2\pi u \frac{k(k+1)}{2N_G}\right), & \text{for } i = 4k + \text{sc\_offset}, \left(\frac{\text{sc\_start} - \text{sc\_offset}}{4}\right) \leq k < \left(\frac{\text{sc\_end} - \text{sc\_offset}}{4}\right) \\ 0 & \text{otherwise} \end{cases}$$

Here, the value of N_G and N_P depend on the FFT size and are specified in Table 217, while the value of u depends on both the FFT size and the cyclic prefix duration and is specified in Table 218. The value of $G_{\text{LOWPAR-GAIN}}$ is beyond the scope of this standard.

Table 217—Specification for the N_G and N_P parameters

N_G	N_P
127	120

Table 218—Specification for the u parameter

N_{CP}	Cyclic prefix duration (μs)	u parameter
1	6.51	17
2	13.02	39
3	19.53	110
4	26.04	112

9.4.1.2.2.2 Forward Other Sector Interference Channel

The Forward Other Sector Interference Channel consists of the last two Orthogonal Frequency Division Multiplexing symbols (i.e., the Orthogonal Frequency Division Multiplexing symbols with indices 6 and 7) in the superframe preamble. These Orthogonal Frequency Division Multiplexing symbols are also known as TDM Pilot 2 and TDM Pilot 3 respectively and are used in the initial acquisition process. In addition, these symbols also carry the other sector interference value that is received from the SFP MAC Protocol. These Orthogonal Frequency Division Multiplexing symbols are first defined as a Walsh code in the time domain. Subsequently, they are converted to the frequency domain using a DFT and modulated onto the Orthogonal Frequency Division Multiplexing subcarriers.

The modulation of TDM Pilots 2 and 3 depends on the value of PilotPhase and on the OSIValue (k_{OSI}), which is received from the SFP MAC Protocol.

9.4.1.2.2.2.1 TDM Pilot 2

First, a time-domain sequence $x(n)$ of length $N_{\text{FFT, TDM Pilot}}$ shall be generated. This sequence is given by the Walsh sequence of length $N_{\text{FFT, TDM Pilot}}$ with index $k_{\text{PILOT}} \bmod N_{\text{FFT, TDM Pilot}}$, where k_{PILOT} denotes the PilotPhase of the sector in the superframe of interest and the notion of a Walsh sequence is defined in 9.2.5.6.

The sequence $x(n)$ shall then be scrambled by a sequence $s(n)$ of length $N_{\text{FFT,TDMPilot}}$, and shall further be multiplied by the complex value $\exp(j \times 2\pi \times k_{\text{OSI}}/3)$ to generate a sequence $y(n)$, i.e., $y(n) = G_{\text{LOWPARGAIN}} A_{\text{PREAMBLE}} x(n) \times s(n) \times \exp(j \times 2\pi \times k_{\text{OSI}}/3)$. The value of $G_{\text{LOWPARGAIN}}$ is beyond the scope of this standard.

The sequence $s(n)$ shall be generated according to the procedure described in 9.2.5.1, with the required 20-bit input seed being given by (starting with the MSB and ending with the LSB) 011010011010111011 a_1a_0 . Here, a_1a_0 is a 2-bit representation of $\lfloor k_{\text{PILOT}}/N_{\text{FFT,TDMPilot}} \rfloor$, with a_1 being the MSB and a_0 being the LSB. Note that a_1a_0 shall always be 0.

The sequence $y(n)$ shall then be converted to the frequency domain by applying a DFT operation of size $N_{\text{FFT,TDMPilot}}$ to generate a sequence $Y(n)$. TDM Pilot 2 shall be generated by modulating the value $Y(i)$ to the subcarrier with index $N_{\text{FFT}}/2 + i - N_{\text{FFT,TDMPilot}}/2$, $0 \leq i < N_{\text{FFT,TDMPilot}}$, if this subcarrier is not a guard subcarrier. All the remaining subcarriers of TDM Pilot 2 shall be unmodulated.

9.4.1.2.2.2 TDM Pilot 3

TDM Pilot 3 shall carry information from the AcqInfo block, which is provided by the SFP Lower MAC protocol. The AcqInfo block has a length of 9 bits, and is interpreted as an integer value k_{SD} between 0 and 511. The 9-bit integer k_{SD} is then mapped to a time-domain sequence $x(n)$ of length $N_{\text{FFT,TDMPilot}}$. This sequence is given by the Walsh sequence of length $N_{\text{FFT,TDMPilot}}$ with index $k_{\text{SD}} \bmod N_{\text{FFT,TDMPilot}}$.

The sequence $x(n)$ shall then be scrambled by a complex sequence $s(n)$ of length $N_{\text{FFT,TDMPilot}}$, and shall further be multiplied by the complex value $\exp(j \times 4\pi \times k_{\text{OSI}}/3)$ to generate a sequence $y(n)$, i.e., $y(n) = G_{\text{LOWPARGAIN}} P_{\text{PREAMBLE}} x(n) \times s(n) \times \exp(j \times 4\pi \times k_{\text{OSI}}/3)$.

The sequence $s(n)$ shall be generated according to the procedure described in 9.2.5.1, with the required 20-bit input seed being given by (starting with the MSB and ending with the LSB) 011010011 $k_8k_7k_6k_5k_4k_3k_2k_1k_0a_1a_0$. Here, $k_8k_7\dots k_0$ is a 9 bit representation of k_{PILOT} , with k_8 being the MSB and k_0 the LSB, while a_1a_0 is a 2-bit representation of $\lfloor k_{\text{SD}}/N_{\text{FFT,TDMPilot}} \rfloor$, with a_1 being the MSB and a_0 being the LSB.

The sequence $y(n)$ shall then be converted to the frequency domain by applying a DFT operation of size $N_{\text{FFT,TDMPilot}}$ to generate a sequence $Y(n)$. TDM Pilot 3 shall be generated by modulating the value $Y(i)$ to the subcarrier with index $N_{\text{FFT}}/2 + i - N_{\text{FFT,TDMPilot}}/2$, $0 \leq i < N_{\text{FFT,TDMPilot}}$, if this subcarrier is not a guard subcarrier. All the remaining subcarriers of TDM Pilot 3 shall be unmodulated.

9.4.1.2.2.3 Forward Preamble Pilot Channel (F-PPICH)

In this subclause, a pilot subcarrier denotes any subcarrier modulated by the Forward Preamble Pilot Channel. The Forward Preamble Pilot Channel shall be present only on Orthogonal Frequency Division Multiplexing symbols with index 1 and 2 within the preamble. Furthermore, the Forward Preamble Pilot Channel shall only modulate subcarriers in the PreamblePilotSubcarrierSet, which is defined as follows:

- $(N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2)$ through $(N_{\text{FFT}}/2 + N_{\text{FFT,TDMPilot}}/2 - 1)$ when EnablePreambleFrequencyReuse is set to 0 and at least one of the following two conditions is true:
 - 1) EnableExpandedQPCH is set to 0.
 - 2) The SuperframeIndex for this superframe is odd.
- 0 through $(N_{\text{FFT}} - 1)$ when EnablePreambleFrequencyReuse is set to 0, EnableExpandedQPCH is set to 1 and the superframe has an even SuperframeIndex.
- $(N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + \text{ReuseIndex} \times N_{\text{FFT,TDMPilot}}/8)$ through $(N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + ((\text{ReuseIndex} + 1) \times N_{\text{FFT,TDMPilot}}/8 - 1))$ when EnablePreambleFrequencyReuse is set to 1, ReuseIndex is defined as $\text{PilotPhase} \bmod 8$, and at least one of the following two conditions is true:

- a) EnableExpandedQPCH is set to 0.
- b) The SuperframeIndex for this superframe is odd.
- $(\text{ReuseIndex} \times N_{\text{FFT}}/8)$ through $((\text{ReuseIndex} + 1) \times N_{\text{FFT}}/8) - 1$ when EnablePreambleFrequencyReuse is set to 1, ReuseIndex is defined as $\text{PilotPhase} \bmod 8$, EnableExpandedQPCH is set to 1 and the superframe has an even SuperframeIndex.

Further define $P_{\text{PPICH}} = N_{\text{FFT}} / \text{NumUsablePreamblePilotSubcarrier}$, where NumUsablePreamblePilotSubcarrier denotes the number of usable subcarriers in the PreamblePilotSubcarrierSet. Also define a quantity $\text{Scr}_{\text{PPICH}}$ according to the following procedure:

- For superframes with an odd SuperframeIndex, let $\text{Scr}_{\text{PPICH}}$ be equal to PilotPhase.
- For superframes with an even value of SuperframeIndex, let $\text{Scr}_{\text{PPICH}}$ be equal to SFNPhase.

The Forward Preamble Pilot Channel shall be modulated on Orthogonal Frequency Division Multiplexing symbol 1 according to the following procedure:

- Generate a complex scrambling sequence s_{PPICH} of length $N_{\text{FFT}}/2$ as described in the common complex scrambling algorithm using seed $f_{\text{PHY-HASH}}(128 \times \text{Scr}_{\text{PPICH}} + 64)$ if the IsSynchronous field of the Overhead Messages Protocol is set to '1', and using seed $f_{\text{PHY-HASH}}(\text{Scr}_{\text{PPICH}} \times 128 + (\text{SuperframeIndex} \bmod 16) \times 4)$ if the IsSynchronous field of the Overhead Messages Protocol is set to '0'.
- The i^{th} value of s_{PPICH} shall be multiplied by $\sqrt{P_{\text{PPICH}}}$ and used to modulate subcarrier $f = (N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + 2i) \bmod N_{\text{FFT}}$ in Orthogonal Frequency Division Multiplexing symbol with index 1 provided
 - 1) Subcarrier f is not a guard subcarrier
 - 2) Subcarrier f is in the PreamblePilotSubcarrierSet

The Forward Preamble Pilot Channel shall be modulated on Orthogonal Frequency Division Multiplexing symbol 2 according to the following procedure:

- Generate a complex scrambling sequence s_{PPICH} of length $N_{\text{FFT}}/2$ as described in the common complex scrambling algorithm using seed $f_{\text{PHY-HASH}}(1 + 128 \times \text{Scr}_{\text{PPICH}} + 64)$ if the IsSynchronous field of the Overhead Messages Protocol is set to '1', and using seed $f_{\text{PHY-HASH}}(\text{Scr}_{\text{PPICH}} \times 128 + (\text{SuperframeIndex} \bmod 16) \times 4)$ if the IsSynchronous field of the Overhead Messages Protocol is set to '0'.
- The i^{th} value of s_{PPICH} shall be used to modulate subcarrier $f = (N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + 2i + 1) \bmod N_{\text{FFT}}$ in Orthogonal Frequency Division Multiplexing symbol with index 2 provided
 - 1) Subcarrier f is not a guard subcarrier and
 - 2) Subcarrier f is in the PreamblePilotSubcarrierSet.

9.4.1.2.2.4 Forward Primary Broadcast Control Channel (F-PBCCH)

The Forward Primary Broadcast Control Channel is carried on the first Orthogonal Frequency Division Multiplexing symbol in the superframe preamble. Each Forward Primary Broadcast Control Channel packet is encoded over one superframe. Forward Primary Broadcast Control Channel modulation depends on the value of EnablePreambleFrequencyReuse, which is a field in the Overhead Messages Protocol.

In this subclause, the quantity $N_{\text{FFT,TDMPilot}}$ is used, which is as defined in 9.4.1.2.2.2.

An Forward Primary Broadcast Channel packet is generated by the SFP MAC Protocol, and is appended with CRC, encoded, channel-interleaved, repeated, data scrambled and modulated according to the procedures described in 9.2.6.1. A CRC length of $N_{\text{CRC,PBCCH}}$ is used while generating the CRC. QPSK modulation is used in the transmission of this channel. A seed equal to $f_{\text{PHY-HASH}}(128 \times p + 64 + 1)$ shall be used for the data scrambling operation if the IsSynchronous field of the Overhead Messages Protocol is set

to '1' and a seed of $f_{\text{PHY-HASH}}(128 \times p + 4 \times (\text{SuperframeIndex} \bmod 16) + 1)$ shall be used otherwise, where p denotes the PilotPhase of the sector in the superframe of interest.

The modulation of the Forward Primary Broadcast Control Channel packet onto subcarriers depends on the value of EnablePreambleFrequencyReuse, which is a field of the Overhead Messages Protocol. If this parameter is set, Forward Primary Broadcast Control Channel transmission from different sectors occupy different subcarrier sets (i.e., frequency reuse is enabled). Otherwise, the Forward Primary Broadcast Control Channel from different sectors occupies the same set of subcarriers and hence interfere with each other.

9.4.1.2.2.4.1 EnablePreambleFrequencyReuse = 0

The i^{th} modulation symbol at the output of the modulator shall be mapped to the subcarrier with index $N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + i$ of the Orthogonal Frequency Division Multiplexing symbol with index 0 in the superframe preamble, if this subcarrier is not a guard subcarrier. Any subcarrier not modulated via the above procedure shall also remain unmodulated by the Forward Primary Broadcast Control Channel. The value of i shall range from 0 to $N_{\text{FFT,TDMPilot}} - 1$.

Each modulation symbol output by the modulator shall further be multiplied by the value $\sqrt{P_{\text{PBCCCH}}} = \sqrt{P_{\text{PREAMBLE}}}$ prior to being mapped to the appropriate subcarrier.

9.4.1.2.2.4.2 EnablePreambleFrequencyReuse = 1

This option is only allowed in Synchronous mode.

In this case, each Forward Primary Broadcast Control Channel packet is modulated only on a subset of subcarriers in the first Orthogonal Frequency Division Multiplexing symbol of the superframe preamble. Different sectors use different sets of subcarriers in order to transmit the Forward Primary Broadcast Control Channel packet. The set of subcarriers used for the transmission of the Forward Primary Broadcast Control Channel is determined by the value of the quantity ReuseIndex, which is defined as PilotPhase mod 8. Note that the value of ReuseIndex changes from superframe to superframe.

The i^{th} modulation symbol at the output of the modulator shall be mapped to the subcarrier with index $N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + \text{ReuseIndex} \times N_{\text{FFT,TDMPilot}}/8 + i$ of the Orthogonal Frequency Division Multiplexing symbol with index 0 in the superframe preamble, if this subcarrier is not a guard subcarrier. Guard subcarriers shall not be modulated. Any subcarrier not modulated via the above procedure shall remain unmodulated by the Forward Primary Broadcast Control Channel. The value of i shall go from 0 to $N_{\text{FFT,TDMPilot}}/8 - 1$.

Further, let n_{sc} denote the number of subcarriers with indices in the range $[N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + \text{ReuseIndex} \times N_{\text{FFT,TDMPilot}}/8, N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + (\text{ReuseIndex} + 1) \times N_{\text{FFT,TDMPilot}}/8 - 1]$, which are not mapped to guard subcarriers.

Each modulation symbol output by the modulator shall further be multiplied by the value $\sqrt{P_{\text{PBCCCH}}} = \sqrt{N_{\text{FFT}}/n_{\text{sc}}}$ prior to being mapped to the appropriate subcarrier.

9.4.1.2.2.5 Forward Secondary Broadcast Control Channel (F-SBCCH)

The Forward Secondary Broadcast Control Channel is carried on the Orthogonal Frequency Division Multiplexing symbols with indices 1 through 4 in the superframe preamble in superframes with an odd value of SuperframeIndex. Each Forward Secondary Broadcast Control Channel packet is encoded over a single superframe.

The modulation of the Forward Secondary Broadcast Control Channel packet onto subcarriers depends on the value of EnablePreambleFrequencyReuse, which is a field of the Overhead Messages Protocol. If this parameter is set, the Forward Secondary Broadcast Control Channel transmission from different sectors occupy different subcarrier sets (i.e., frequency reuse is enabled). Otherwise, the Forward Secondary Broadcast Control Channel from different sectors occupies the same set of subcarriers and hence interfere with each other.

In this subclause, the quantity $N_{\text{FFT,TDMPilot}}$ is used, which is defined in 9.4.1.2.2.2.

A Forward Secondary Broadcast Channel packet is generated by the SFP MAC Protocol, and is appended with CRC, encoded, channel-interleaved, repeated, data scrambled and modulated according to the procedures described in 9.2.6.1. A CRC length of $N_{\text{CRC,SBCCH}}$ is used while generating the CRC. QPSK modulation is used in the transmission of this channel. A seed equal to $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 64 + 2)$ shall be used for the data scrambling operation if the IsSynchronous field of the Overhead Messages Protocol is set to '1' and a seed of $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 4 \times (\text{SuperframeIndex mod } 16) + 2)$ shall be used otherwise, where p denotes the PilotPhase of the sector in the superframe of interest, and s denotes the quantity SBCCHScramblingSeed, which is computed by the Overhead Messages Protocol.

9.4.1.2.2.5.1 EnablePreambleFrequencyReuse = 0

The i^{th} modulation symbol at the output of the modulator shall be multiplied by $\sqrt{P_{\text{SBCCH}}}$ and mapped to the subcarrier with index $N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + (i \bmod N_{\text{FFT,TDMPilot}})$ of the Orthogonal Frequency Division Multiplexing symbol with index $\lfloor i / N_{\text{FFT,TDMPilot}} \rfloor + 1$ in the superframe preamble, if this subcarrier is a usable subcarrier and is additionally not a pilot subcarrier. $P_{\text{SBCCH}} = P_{\text{PPICH}}$ is the power density to be used for modulating the Forward Preamble Pilot Channel in the same superframe, and is defined in 9.4.1.2.2.3. Any subcarrier not modulated via the above procedure shall remain unmodulated by the Forward Secondary Broadcast Control Channel. The value of i shall range from 0 to $4 \times N_{\text{FFT,TDMPilot}} - 1$.

9.4.1.2.2.5.2 EnablePreambleFrequencyReuse = 1

This option is only allowed in Synchronous mode.

In this case, each Forward Secondary Broadcast Control Channel packet is modulated only on a subset of subcarriers in the Orthogonal Frequency Division Multiplexing symbols with indices 1 through 4 of the superframe preamble. Different sectors use different sets of subcarriers in order to transmit the Forward Secondary Broadcast Control Channel packet. The set of subcarriers used for the transmission of Forward Secondary Broadcast Control Channel is determined by the value of the quantity ReuseIndex, which is defined as PilotPhase mod 8.

The i^{th} modulation symbol at the output of the modulator shall be multiplied by $\sqrt{P_{\text{SBCCH}}}$ and mapped to the subcarrier with index $N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + \text{ReuseIndex} \times N_{\text{FFT,TDMPilot}}/8 + (i \bmod N_{\text{FFT,TDMPilot}}/8)$ of the Orthogonal Frequency Division Multiplexing symbol with index $\lfloor 8 \times i / N_{\text{FFT,TDMPilot}} \rfloor + 1$ in the superframe preamble, if this subcarrier is a usable subcarrier and is not a pilot subcarrier. $P_{\text{SBCCH}} = P_{\text{PPICH}}$ is the power density to be used for modulating the Forward Preamble Pilot Channel in that superframe, and is defined in 9.4.1.2.2.3. Any subcarrier not modulated via the above procedure shall remain unmodulated by the Forward Secondary Broadcast Control Channel. The value of i shall range from 0 to $(N_{\text{FFT,TDMPilot}})/2 - 1$.

9.4.1.2.2.6 Forward Quick Paging Channel

The Forward Quick Paging Channel (F-QPCH) is carried on the Orthogonal Frequency Division Multiplexing symbols with indices 1 through 4 in the superframe preamble in alternate superframes. The

Forward Quick Paging Channel shall be carried on superframes with an even value of SuperframeIndex. Each Forward Quick Paging Channel packet is encoded over a single superframe.

The modulation of the Forward Quick Paging Channel packet onto subcarriers depends on the value of EnablePreambleFrequencyReuse, which is a field of the Overhead Messages Protocol. If this parameter is set to 1, Forward Quick Paging Channel transmission from different sectors occupy different subcarrier sets (i.e., frequency reuse is enabled). Otherwise, the Forward Quick Paging Channel from different sectors occupies the same set of subcarriers and hence interfere with each other.

Forward Quick Paging Channel modulation also depends on the value of EnableExpandedQPCH, which is a field of the Overhead Messages Protocol. This variable determines how many Forward Quick Paging Channel packets are transmitted in a single superframe.

EnableExpandedQPCH may not be set to 1 unless $\lfloor (N_{\text{FFT}} - N_{\text{GUARD}} + 128) / 512 \rfloor > 1$. If EnableExpandedQPCH is set to 0, then a single Forward Quick Paging Channel packet shall be transmitted in each superframe preamble containing the Forward Quick Paging Channel. If EnableExpandedQPCH is set to 1, then the number of Forward Quick Paging Channel packets transmitted in each superframe preamble shall be given by $\lfloor (N_{\text{FFT}} - N_{\text{GUARD}} + 128) / 512 \rfloor$.

In this subclause, the quantity $N_{\text{FFT,TDMPilot}}$ is used, which is defined in 9.4.1.2.2.2.

9.4.1.2.2.6.1 EnablePreambleFrequencyReuse = 0 and EnableExpandedQPCH = 0

In this case, each Forward Quick Paging Channel packet is generated by the SFP MAC Protocol, and is appended with CRC, encoded, channel-interleaved, repeated, data scrambled and modulated according to the procedures described in 9.2.6.1. A CRC length of $N_{\text{CRC,QPCH}}$ is used while generating the CRC. QPSK modulation is used in the transmission of this channel. A seed equal to $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 64 + 3)$ shall be used for the data scrambling operation if the IsSynchronous field of the Overhead Messages Protocol is set to '1' and a seed of $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 4 \times (\text{SuperframeIndex mod } 16) + 3)$ shall be used otherwise, where p denotes the SFNPhase of the sector in the superframe of interest, and s denotes the quantity QPCHScramblingSeed, which is computed by the Overhead Messages Protocol for each superframe.

The i^{th} modulation symbol at the output of the modulator shall be multiplied by $\sqrt{P_{\text{QPCH}}}$ and mapped to the subcarrier with index $N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + (i \text{ mod } N_{\text{FFT,TDMPilot}})$ of the Orthogonal Frequency Division Multiplexing symbol with index $\lfloor i / N_{\text{FFT,TDMPilot}} \rfloor + 1$ in the superframe preamble, if this subcarrier is a usable subcarrier and is additionally not a pilot subcarrier. $P_{\text{QPCH}} = P_{\text{PPICH}}$ is the power density to be used for modulating the Forward Preamble Pilot Channel in the same superframe, and is defined in 9.4.1.2.2.3. Any subcarrier not modulated via the above procedure shall remain unmodulated by the Forward Quick Paging Channel. The value of i ranges from 0 to $4 \times N_{\text{FFT,TDMPilot}} - 1$.

9.4.1.2.2.6.2 EnablePreambleFrequencyReuse = 1 and EnableExpandedQPCH = 0

In this case, each Forward Quick Paging Channel packet is modulated only on a subset of subcarriers in the Orthogonal Frequency Division Multiplexing symbols with indices 1 through 4 of the superframe preamble. Different sectors use different sets of subcarriers in order to transmit the Forward Quick Paging Channel packet. The set of subcarriers used for the transmission of the Forward Quick Paging Channel is determined by the value of the quantity ReuseIndex, which is defined as PilotPhase mod 8.

A Forward Quick Paging Channel packet is generated by the SFP MAC Protocol, and is appended with CRC, encoded, channel-interleaved, repeated and modulated according to the procedures described in 9.2.6. A CRC length of $N_{\text{CRC,QPCH}}$ is used while generating the CRC. QPSK modulation is used in the transmission of this channel. A seed equal to $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 64 + 3)$ shall be used for the data scrambling operation if the IsSynchronous field of the Overhead Messages Protocol is set to '1' and a seed of

$f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 4 \times (\text{SuperframeIndex mod } 16) + 3)$ shall be used otherwise, where p denotes the SFNPhase of the sector in the superframe of interest, and s denotes the quantity QPCHScramblingSeed, which is computed by the Overhead Messages Protocol for each superframe.

The i^{th} modulation symbol at the output of the modulator shall be multiplied by $\sqrt{P_{\text{QPCH}}}$ and mapped to the subcarrier with index $N_{\text{FFT}}/2 - N_{\text{FFT,TDMPilot}}/2 + (\text{ReuseIndex} \times N_{\text{FFT,TDMPilot}}/8) + (i \bmod N_{\text{FFT,TDMPilot}}/8)$ of the Orthogonal Frequency Division Multiplexing symbol with index $\lfloor 8 \times i/N_{\text{FFT,TDMPilot}} \rfloor + 1$ in the superframe preamble, if this subcarrier is a usable subcarrier and is not a pilot subcarrier. $P_{\text{QPCH}} = P_{\text{PPICH}}$ is the power density to be used for modulating the Forward Preamble Pilot Channel in the same superframe, and is defined in 9.4.1.2.2.3. Any subcarrier not modulated via the above procedure shall remain unmodulated by the Forward Quick Paging Channel. The value of i ranges from 0 to $(N_{\text{FFT,TDMPilot}})/2 - 1$.

9.4.1.2.2.6.3 EnablePreambleFrequencyReuse = 0 and EnableExpandedQPCH = 1

In this case, each Forward Quick Paging Channel packet is generated by the SFP MAC Protocol, and is appended with CRC, encoded, channel-interleaved, repeated and modulated according to the procedures described in 9.2.6.1. The SFP MAC also provides an indexing of the different Forward Quick Paging Channel packets generated in the same superframe, with the indexing being from 0 through $\lfloor (N_{\text{FFT}} - N_{\text{GUARD}} + 128)/512 \rfloor - 1$. A CRC length of $N_{\text{CRC,QPCH}}$ is used while generating the CRC. QPSK modulation is used in the transmission of this channel. A seed equal to $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 64 + 3)$ shall be used for the data scrambling operation if the IsSynchronous field of the Overhead Messages Protocol is set to '1' and a seed of $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 4 \times (\text{SuperframeIndex mod } 16) + 3)$ shall be used otherwise, where p denotes the SFNPhase of the sector in the superframe of interest, and s denotes the quantity QPCHScramblingSeed, which is computed by the Overhead Messages Protocol for each superframe.

The i^{th} modulation symbol at the output of the modulator of the k^{th} packet shall be multiplied by $\sqrt{P_{\text{QPCH}}}$ and mapped to the subcarrier with index $512 \times k + (i \bmod 512)$ of the Orthogonal Frequency Division Multiplexing symbol with index $\lfloor i/512 \rfloor + 1$ in the superframe preamble, if this subcarrier is a usable subcarrier and is additionally not a pilot subcarrier. $P_{\text{QPCH}} = P_{\text{PPICH}}$ is the power density to be used for modulating the Forward Preamble Pilot Channel in the same superframe, and is defined in 9.4.1.2.2.3. Any subcarrier not modulated via the above procedure shall remain unmodulated by the Forward Quick Paging Channel. The value of i ranges from 0 to $4 \times N_{\text{FFT,TDMPilot}} - 1$.

9.4.1.2.2.6.4 EnablePreambleFrequencyReuse = 1 and EnableExpandedQPCH = 1

This option is only allowed in Synchronous mode.

In this case, each Forward Quick Paging Channel packet is modulated only on a subset of subcarriers in the Orthogonal Frequency Division Multiplexing symbols with indices 1 through 4 of the superframe preamble. Different sectors use different sets of subcarriers in order to transmit the Forward Quick Paging Channel packet. The set of subcarriers used for the transmission of Forward Quick Paging Channel is determined by the value of the quantity ReuseIndex, which is defined as $\text{PilotPhase mod } 8$.

Each Forward Quick Paging Channel packet is generated by the SFP MAC Protocol, and is appended with CRC, encoded, channel-interleaved, repeated and modulated according to the procedures described in 9.2.6.1. The SFP MAC also provides an indexing of the different Forward Quick Paging Channel packets generated in the same superframe, where the index $k = 0, 1, \dots, \lfloor (N_{\text{FFT}} - N_{\text{GUARD}} + 1)/512 \rfloor - 1$. A CRC length of $N_{\text{CRC,QPCH}}$ is used while generating the CRC. QPSK modulation is used in the transmission of this channel. A seed equal to $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 64 + 3)$ shall be used for the data scrambling operation if the IsSynchronous field of the Overhead Messages Protocol is set to '1' and a seed of $f_{\text{PHY-HASH}}(2^{16} \times s + 2^7 \times p + 4 \times (\text{SuperframeIndex mod } 16) + 3)$ shall be used otherwise, where p denotes the SFNPhase of the sector in the superframe of interest, and s denotes the quantity QPCHScramblingSeed, which is computed by the Overhead Messages Protocol for each superframe multiplied by $\sqrt{P_{\text{QPCH}}}$.

The i^{th} modulation symbol at the output of the modulator of the k^{th} packet shall be mapped to the subcarrier with index $k \times 512 + \text{ReuseIndex} \times 512/8 + (i \bmod (512/8))$ of the Orthogonal Frequency Division Multiplexing symbol with index $\lfloor 8 \times i/512 \rfloor + 1$ in the superframe preamble, if this subcarrier is a usable subcarrier and is not a pilot subcarrier. $P_{\text{QPCH}} = P_{\text{PPICH}}$ is the power density to be used for modulating the Forward Preamble Pilot Channel in the same superframe, and is defined in 9.4.1.2.2.3. Any subcarrier not modulated via the above procedure shall remain unmodulated by the Forward Quick Paging Channel. The value of i ranges from 0 to $(N_{\text{FFT,TDMPILOT}})/2 - 1$.

9.4.1.2.3 Pilot Channels

This subclause describes the pilot channels that are present in the Forward Link PHY frames. It does not include the Forward Preamble Pilot Channel, which is present in the superframe preamble.

The pilot channels in the Forward Link PHY frames consist of the Forward Common Pilot Channel, the Forward Dedicated Pilot Channel, the Forward Channel Quality Indicator Pilot Channel and the Forward Beacon Pilot Channel. The structure of the pilot channels depends on the value of the parameter ResourceChannelMuxMode, which is a field of the Overhead Messages Protocol.

When ResourceChannelMuxMode = 1, the Forward Common Pilot Channel is present in every Forward Link PHY frame and spans the entire usable bandwidth. The Forward Common Pilot Channel is designed to be used as a channel estimation pilot in this case. The Forward Dedicated Pilot Channel and the Forward Channel Quality Indicator Pilot Channel are absent in this case.

When ResourceChannelMuxMode = 2, the Forward Common Pilot Channel is transmitted in every Forward Link PHY frame over the DRCH subzones. The Forward Dedicated Pilot Channel is transmitted in every Forward Link PHY frame over the BRCH subzones. The Forward Common Pilot Channel and the Forward Dedicated Pilot Channel are designed to be used for channel estimation in DRCH and BRCH zones respectively. Note that the Forward Common Pilot Channel shall be completely absent if there are no DRCH subzones. Similarly, the Forward Dedicated Pilot Channel shall be completely absent if there are no BRCH subzones. Finally, the Forward Channel Quality Indicator Pilot Channel is a low overhead pilot channel that is transmitted in one out of every eight Forward Link PHY frames, and is designed to be used by the access terminal to measure channel quality and to support precoding.

9.4.1.2.3.1 Forward Common Pilot Channel

The Forward Common Pilot Channel (F-CPICH) provides a wideband reference of the channel across the whole band.

9.4.1.2.3.1.1 Forward Common Pilot Channel subcarriers

The Forward Common Pilot Channel shall be transmitted from disjoint sets of subcarriers from each of NumEffectiveAntennas effective antennas, where NumEffectiveAntennas is a parameter of the Overhead Messages Protocol and takes values between 1 and 4. These antennas are indexed from 0 to NumEffectiveAntennas – 1. For each Forward Link PHY frame, define a quantity CommonPilotFreqInterlace taking values from 0 to 15 according to the following procedure:

- If CPICHHoppingMode, which is a field of the Overhead Messages Protocol, takes the value “Random” then CommonPilotFreqInterlace shall be set to output modulo CommonPilotSpacing of the hash function defined in 9.2.5.4, with input given by $2^{20} \times \lfloor j/4 \rfloor + \text{SectorSeed}$, where j denotes the index of the OFDM Symbol within the Forward Link PHY frame, SectorSeed is defined in 9.2.2.2.3, and CommonPilotSpacing is a field of the Overhead Messages Protocol.
- If CPICHHoppingMode takes the value “Deterministic,” then CommonPilotFrequencyInterlace shall be set to $\text{PilotPN} \bmod \text{CommonPilotSpacing}$, if $0 \leq j < 4$, and shall be set to $(\text{PilotPN} + \text{CommonPilotSpacing}/4) \bmod \text{CommonPilotSpacing}$ otherwise. Here CommonPilotSpacing is a

field of the Overhead Messages Protocol, and j denotes the index of the OFDM symbol within the PHY frame.

In each Forward Link PHY frame, a usable subcarrier with index i in the Orthogonal Frequency Division Multiplexing symbol with index j within the Forward Link PHY frame carries the Forward Common Pilot Channel from effective antenna 0 if the following conditions are satisfied:

- Either ResourceChannelMuxMode = 1, or ResourceChannelMuxMode = 2 and the subcarrier with index i belongs to a DRCH subzone.
- The subcarrier with index i in the Orthogonal Frequency Division Multiplexing symbol with index j is not occupied by the Forward Channel Quality Indicator Pilot Channel. See 9.4.1.2.3.3 for the details of Forward Channel Quality Indicator Pilot Channel modulation.
- One of the following four conditions is satisfied:
 - a) $j = 0$ and $i \bmod \text{CommonPilotSpacing} = \text{CommonPilotFreqInterlace} \bmod \text{CommonPilotSpacing}$.
 - b) $j = 1$ and $i \bmod \text{CommonPilotSpacing} = (\text{CommonPilotFreqInterlace} + \text{CommonPilotSpacing}/2) \bmod \text{CommonPilotSpacing}$.
 - c) $j = 4$ and $i \bmod \text{CommonPilotSpacing} = \text{CommonPilotFreqInterlace} \bmod \text{CommonPilotSpacing}$.
 - d) $j = 5$ and $i \bmod \text{CommonPilotSpacing} = (\text{CommonPilotFreqInterlace} + \text{CommonPilotSpacing}/2) \bmod \text{CommonPilotSpacing}$.

In each Forward Link PHY frame, a usable subcarrier with index i in the Orthogonal Frequency Division Multiplexing symbol with index j within the Forward Link PHY frame carries the Forward Common Pilot Channel from antenna 1 if the following conditions are satisfied:

- NumEffectiveAntennas ≥ 2 .
- Either ResourceChannelMuxMode = 1, or ResourceChannelMuxMode = 2 and the subcarrier with index i belongs to a DRCH subzone.
- The subcarrier with index i in the Orthogonal Frequency Division Multiplexing symbol with index j is not occupied by the Forward Channel Quality Indicator Pilot Channel. See 9.4.1.2.3.3 for the details of the Forward Channel Quality Indicator Pilot Channel modulation.
- One of the following four conditions is satisfied:
 - a) $j = 0$ and $i \bmod \text{CommonPilotSpacing} = (\text{CommonPilotFreqInterlace} + \text{CommonPilotSpacing}/2) \bmod \text{CommonPilotSpacing}$.
 - b) $j = 1$ and $i \bmod \text{CommonPilotSpacing} = \text{CommonPilotFreqInterlace} \bmod \text{CommonPilotSpacing}$.
 - c) $j = 4$ and $i \bmod \text{CommonPilotSpacing} = (\text{CommonPilotFreqInterlace} + \text{CommonPilotSpacing}/2) \bmod \text{CommonPilotSpacing}$.
 - d) $j = 5$ and $i \bmod \text{CommonPilotSpacing} = \text{CommonPilotFreqInterlace} \bmod \text{CommonPilotSpacing}$.

In each Forward Link PHY frame, a usable subcarrier with index i in the Orthogonal Frequency Division Multiplexing symbol with index j within the Forward Link PHY frame carries the Forward Common Pilot Channel from antenna 2 if the following conditions are satisfied:

- NumEffectiveAntennas ≥ 3 .
- Either ResourceChannelMuxMode = 1, or ResourceChannelMuxMode = 2 and the subcarrier with index i belongs to a DRCH subzone.
- The subcarrier with index i in the Orthogonal Frequency Division Multiplexing symbol with index j is not occupied by the Forward Channel Quality Indicator Pilot Channel.
- One of the following four conditions is satisfied:

- a) $j = 2$ and $i \bmod \text{CommonPilotSpacing} = \text{CommonPilotFreqInterlace} \bmod \text{CommonPilotSpacing}$.
- b) $j = 3$ and $i \bmod \text{CommonPilotSpacing} = (\text{CommonPilotFreqInterlace} + \text{CommonPilotSpacing}/2) \bmod \text{CommonPilotSpacing}$.
- c) $j = 6$ and $i \bmod \text{CommonPilotSpacing} = \text{CommonPilotFreqInterlace} \bmod \text{CommonPilotSpacing}$.
- d) $j = 7$ and $i \bmod \text{CommonPilotSpacing} = (\text{CommonPilotFreqInterlace} + \text{CommonPilotSpacing}/2) \bmod \text{CommonPilotSpacing}$.

In each Forward Link PHY frame, a usable subcarrier with index i in the Orthogonal Frequency Division Multiplexing symbol with index j within the Forward Link PHY frame carries the Forward Common Pilot Channel from antenna 3 if the following conditions are satisfied:

- $\text{NumEffectiveAntennas} = 4$.
- Either $\text{ResourceChannelMuxMode} = 1$, or $\text{ResourceChannelMuxMode} = 2$ and the subcarrier with index i belongs to a DRCH subzone.
- The subcarrier with index i in the Orthogonal Frequency Division Multiplexing symbol with index j is not occupied by the Forward Channel Quality Indicator Pilot Channel. See 9.4.1.2.3.3 for the details of Forward Channel Quality Indicator Pilot Channel modulation.
- One of the following four conditions is satisfied:
 - a) $j = 2$ and $i \bmod \text{CommonPilotSpacing} = (\text{CommonPilotFreqInterlace} + \text{CommonPilotSpacing}/2) \bmod \text{CommonPilotSpacing}$.
 - b) $j = 3$ and $i \bmod \text{CommonPilotSpacing} = \text{CommonPilotFreqInterlace} \bmod \text{CommonPilotSpacing}$.
 - c) $j = 6$ and $i \bmod \text{CommonPilotSpacing} = (\text{CommonPilotFreqInterlace} + \text{CommonPilotSpacing}/2) \bmod \text{CommonPilotSpacing}$.
 - d) $j = 7$ and $i \bmod \text{CommonPilotSpacing} = \text{CommonPilotFreqInterlace} \bmod \text{CommonPilotSpacing}$.

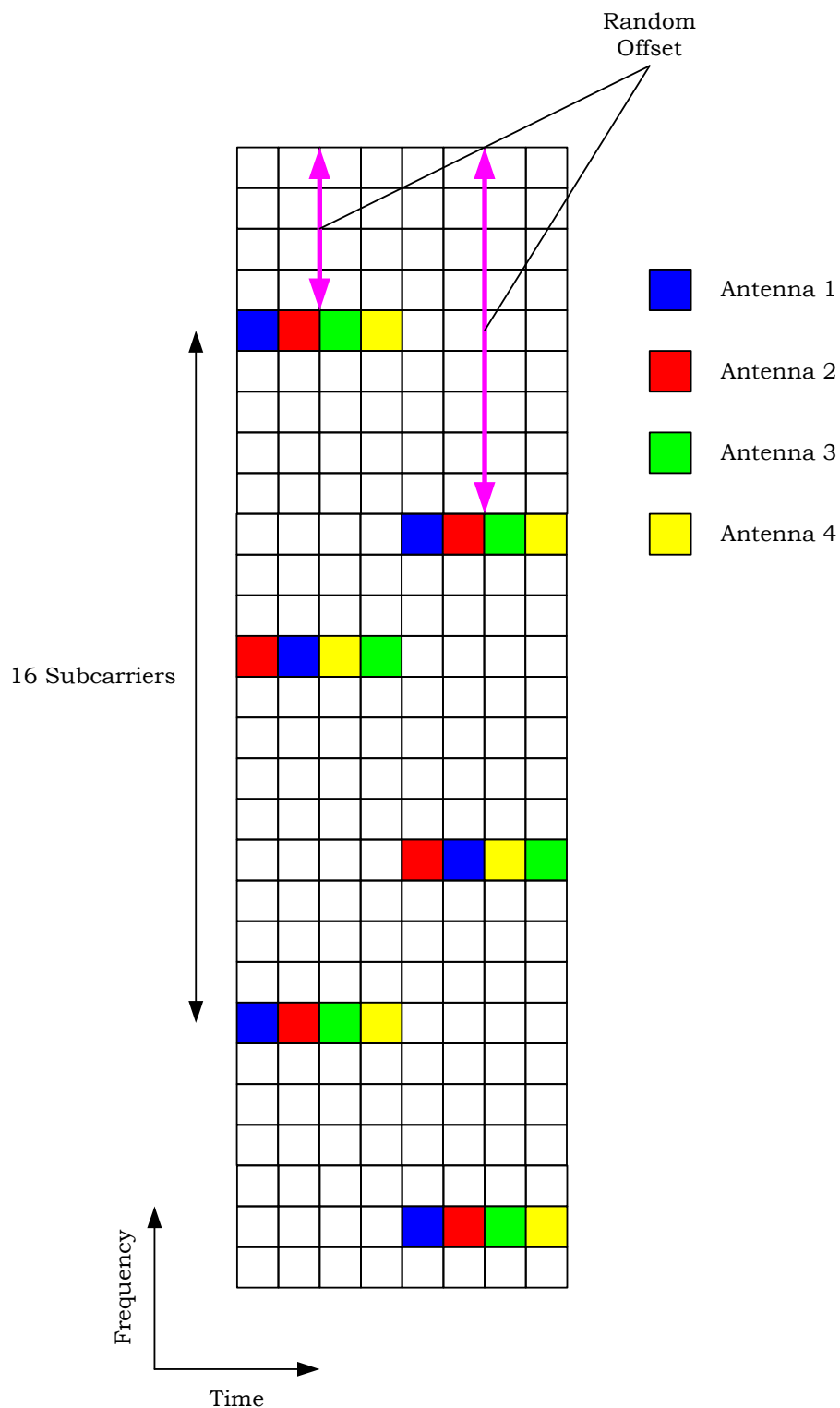


Figure 133—An example of Forward Common Pilot Channel placement for the case where CPICHoppingMode takes the value 'Random' and NumEffectiveAntennas = 4

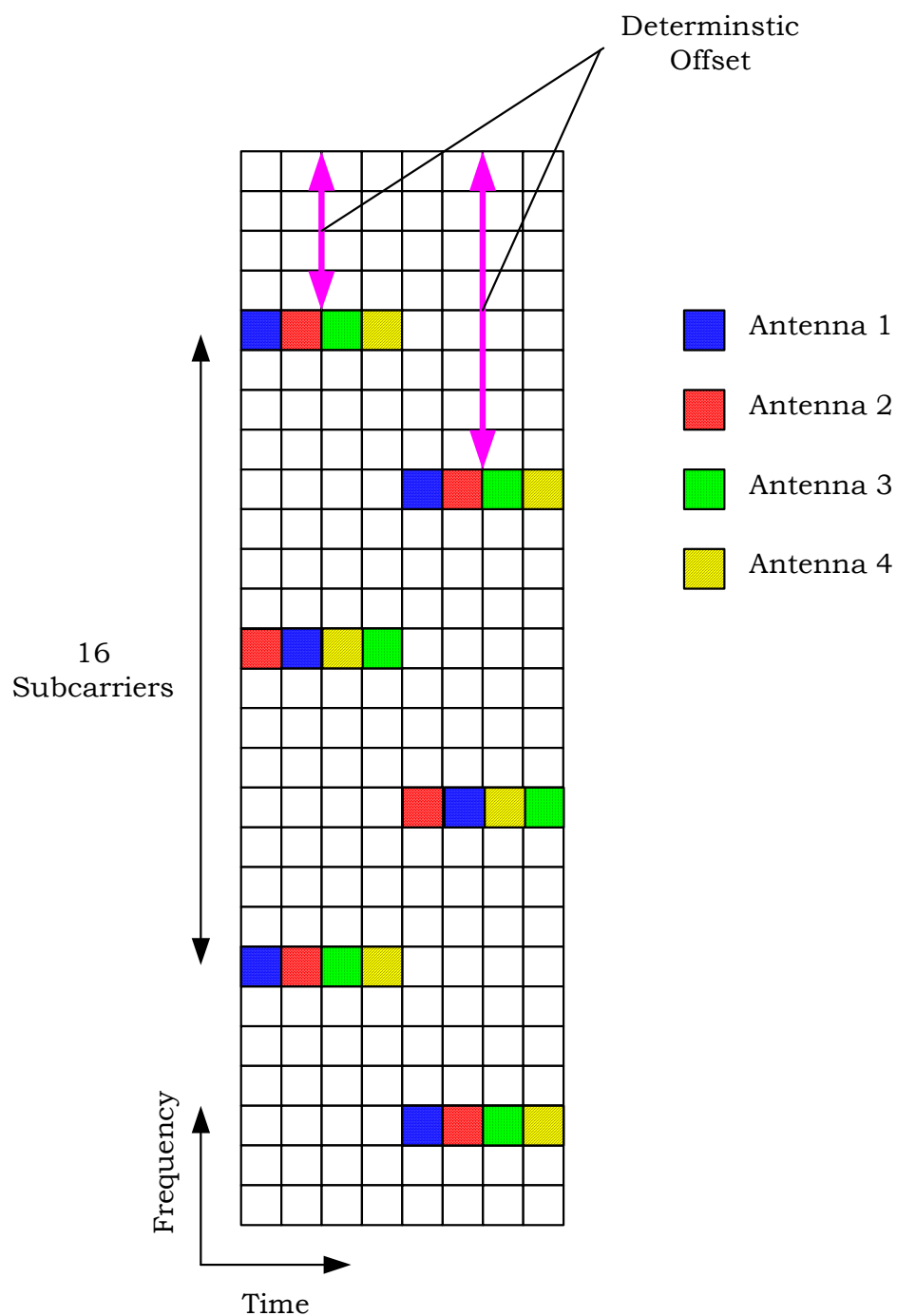


Figure 134—An example of Forward Common Pilot Channel placement for the case where CPICHoppingMode takes the value 'Deterministic' and NumCommonPilotTransmitAntennas = 4

9.4.1.2.3.1.2 Forward Common Pilot Channel Value

Define a Forward Common Pilot Channel index for each subcarrier occupied by the Forward Common Pilot Channel according to the following procedure:

- a) At the beginning of each frame, initialize an Orthogonal Frequency Division Multiplexing symbol counter i to 0, a subcarrier counter j to 0 and a Forward Common Pilot Channel index counter k to 0.
- b) If the subcarrier j in Orthogonal Frequency Division Multiplexing symbol i in the superframe is occupied by the Forward Common Pilot Channel from any effective antenna, then:
 - c) Set its Forward Common Pilot Channel index to k .
 - d) increment k by 1
- e) Increment j by 1. If $j = N_{\text{FFT}}$, set j to 0 and increment i by 1.
- f) Repeat steps b) and c) until $i = N_{\text{FRAME}}$.

A subcarrier that is occupied by the Forward Common Pilot Channel from any effective antenna with index a shall be modulated with the complex value $(\sqrt{P}, 0)$ times the k^{th} entry of a scrambling sequence S from that effective antenna. Other antennas shall not transmit any signal on this subcarrier. The value of P is given by the field `CommonPilotTransmitPower` field of the Overhead Messages Protocol. The value of k is given by the Forward Common Pilot Channel index of the subcarrier which is defined above. The scrambling sequence shall be as specified in 9.2.5.2, with the seed being given by the output of the hash function defined in 9.2.5.4, with the input to the hash function given by $(40 \times 2^{20} \times 8 + 8 \times \text{SectorSeed} + a \bmod 8)$, where `SectorSeed` is as defined in 9.2.2.2.3.

9.4.1.2.3.2 Forward Dedicated Pilot Channel

The Forward Dedicated Pilot Channel (F-DPICH) shall be present only when `ResourceChannelMuxMode` = 2. In this mode, the Forward Dedicated Pilot Channel is present in BRCH subzones. As described in 9.2.14.2, the BRCH subzone is divided into units of hop-port blocks. Each hop-port block consists of 16 contiguous hop-ports, which are mapped by the hopping permutation to a contiguous set of subcarriers. Also, the set of subcarriers corresponding to a hop-port block does not change over one PHY frame. (Note however that since the Forward Link supports SDMA, a two hop-port blocks can be mapped to the same set of subcarriers.) Therefore, the set of resources (over time and frequency) in a BRCH subzone can be divided into units of tiles, where a tile is a contiguous 16x8 rectangle of hop-ports (16 in frequency and 8 in time) which are mapped to a contiguous 16x8 rectangle of subcarriers (16 in frequency and 8 in time).

Each tile in a BRCH subzone can be assigned to the control segment, to the Forward Data Channel, or can be left blank. The Forward Dedicated Pilot Channel shall be present in each tile in a BRCH subzone, i.e., some of the subcarriers in each tile shall be designated as Forward Dedicated Pilot Channel subcarriers. Each tile in a BRCH subzone may be transmitted from up to four tile-antennas, where a tile-antenna is defined in Clause 3. The Forward Dedicated Pilot Channel waveform shall be defined separately from each of these tile-antennas. The tile-antennas used to transmit the Forward Dedicated Pilot Channel in a tile shall be the same as the tile-antennas used to transmit the control segment or the Forward Data Channel from that tile. If two tiles map to the same frequency resources, then the Forward Dedicated Pilot Channel waveforms assigned to these tiles shall be superimposed. The Forward Dedicated Pilot Channel configuration in each tile depends on the following parameters:

- The number of tile-antennas n_t : n_t is equal to 1 if the tile is occupied by the control segment. If the tile is occupied by the Forward Data Channel, the value of n_t is the same as the number of tile-antennas used to transmit the Forward Data Channel from that tile.
- Forward Dedicated Pilot Channel format: The Forward Dedicated Pilot Channel format can take one of three values, indexed 0, 1, and 2. Forward Dedicated Pilot Channel format 0 shall be used for tiles occupied by the control segment. For tiles occupied by the Forward Data Channel, the Forward Dedicated Pilot Channel format depends on the Forward Data Channel assignment occupying the tile, and is determined by the FTC MAC Protocol.

- Energy per modulation symbol: This quantity, denoted by P , is defined separately for each tile-antenna and each tile, but is fixed for all the modulation symbols from the same tile-antenna within a tile. For tiles in the control segment, the exact procedure determination of P is outside the scope of this standard. The power density used to transmit the Forward Dedicated Pilot Channel in a given tile is equal to the power density used to transmit the Forward Data Channel in that tile.
- CodeOffset: This is an integer between 0 and 3. It takes value 0 for tiles belonging to the Forward Link Control segment. For tiles belonging to the Forward Data Channel, the value is determined by the value of SubtreeIndex for that Forward Data Channel assignment, which is determined by the FTC MAC Protocol. If SubtreeIndex is equal to zero, then CodeOffset is equal to zero. For other values of SubtreeIndex, the value of CodeOffset is given by $\text{FLDPICHCodeOffset}_{\text{SubtreeIndex}}$, which is a field of the Overhead Messages Protocol. The access network shall ensure that the same pilot formats are used for different assignments (on different subtrees) that map to any single tile.

The locations of the Forward Dedicated Pilot Channel subcarriers in a tile depend on the Forward Dedicated Pilot Channel format and are shown in Figure 135. Note that the hop-ports within a tile are indexed 0 to 15 in increasing order of hop-port index, and the Orthogonal Frequency Division Multiplexing symbols within a Forward Link PHY frame are indexed 0 to 7 with the earliest Orthogonal Frequency Division Multiplexing symbol being indexed 0.

The indexing of the Forward Dedicated Pilot Channel subcarriers are as described above whether or not the Forward Dedicated Pilot Channel is transmitted on these subcarriers.

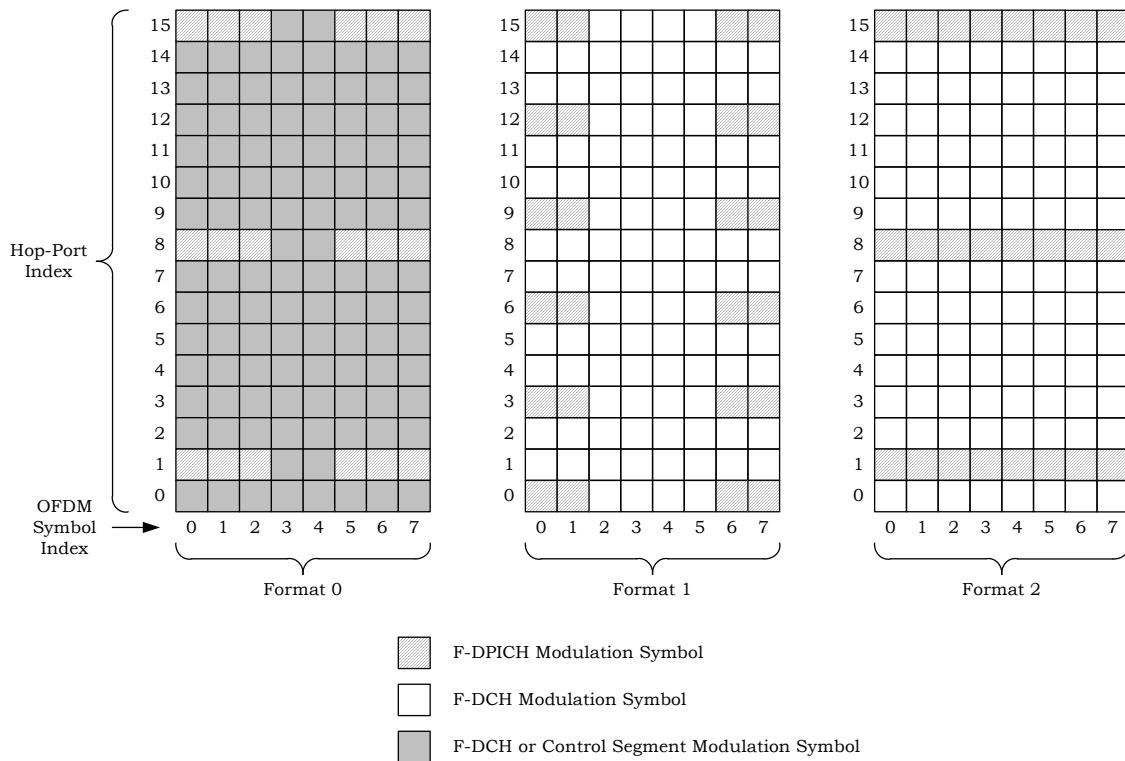


Figure 135—Location of Forward Dedicated Pilot Channel Subcarriers within a tile for the Different Forward Dedicated Pilot Channel formats

9.4.1.2.3.2.1 Forward Dedicated Pilot Channel Format 0

For Forward Dedicated Pilot Channel Format 0, the Forward Dedicated Pilot Channel shall occupy the modulation symbol of the tile if the hop-port index within the tile is in the set $\{1, 8, 15\}$ and the Orthogonal

Frequency Division Multiplexing symbol index t within the Forward Link PHY frame is in the set $T = \{0, 1, 2, 5, 6, 7\}$.

The complex value of the Forward Dedicated Pilot Channel modulation symbol on the tile-antenna with index k shall depend only on the Orthogonal Frequency Division Multiplexing symbol index t and shall be given by

$$S_{t,k} = \sqrt{P} \exp \left(\frac{j2\pi}{3}(k + \text{CodeOffset})t \right), \text{ if } t < 4, \text{ and}$$

$$S_{t,k} = \sqrt{P} \exp \left(\frac{j2\pi}{3}(k + \text{CodeOffset})(7 - t) \right), \text{ if } t \geq 4;$$

where j denotes the complex number (0, 1) and P denotes the energy per modulation symbol on tile-antenna k used by the Forward Dedicated Pilot Channel.

9.4.1.2.3.2 Forward Dedicated Pilot Channel Format 1

For Forward Dedicated Pilot Channel Format 1, the Forward Dedicated Pilot Channel shall occupy the modulation symbol of the tile if the hop-port index within the tile is in the set $\{0, 3, 6, 9, 12, 15\}$ and the Orthogonal Frequency Division Multiplexing symbol index, t , is in the set $T = \{0, 1, 6, 7\}$.

The complex value of the Forward Dedicated Pilot Channel modulation symbol on the tile antenna with index k shall be given by

$$S_{t,k} = \sqrt{P} \exp (j\pi(k + \text{CodeOffset})t), \text{ if } t < 4, \text{ and}$$

$$S_{t,k} = \sqrt{P} \exp (j\pi(k + \text{CodeOffset})(7 - t)), \text{ if } t \geq 4;$$

where j denotes the complex number (0, 1), and P denotes the energy per modulation symbol on tile-antenna k used by the Forward Dedicated Pilot Channel.

9.4.1.2.3.2.3 Forward Dedicated Pilot Channel Format 2

For Forward Dedicated Pilot Channel Format 2, the Forward Dedicated Pilot Channel shall occupy the modulation symbol of the tile if the hop-port index is in the set $\{1, 8, 15\}$ and the Orthogonal Frequency Division Multiplexing symbol index, t , is in the set $\{0, 1, 2, 3, 4, 5, 6, 7\}$. The complex value of the Forward Dedicated Pilot Channel modulation symbol on the tile antenna with index k (using the same symbols as 9.4.1.2.2.3) shall be given by

$$S_{t,k} = \sqrt{P} \exp \left(\frac{j\pi}{2}(k + \text{CodeOffset})t \right), \text{ if } t < 4, \text{ and}$$

$$S_{t,k} = \sqrt{P} \exp \left(\frac{j\pi}{2}(k + \text{CodeOffset})(7 - t) \right), \text{ if } t \geq 4;$$

No Forward Dedicated Pilot Channel shall be transmitted on Orthogonal Frequency Division Multiplexing symbol t if it is a BeaconOnlyOFDMSymbol.

9.4.1.2.3.2.4 Forward Dedicated Pilot Channel Scrambling

9.4.1.2.3.2.4.1 Forward Dedicated Pilot Channel Index Definition

Forward Dedicated Pilot Channel scrambling is done on a tile-by-tile basis. The scrambling symbols that shall be used shall be those generated for subcarriers that correspond to the Forward Dedicated Pilot Channel hop-ports (via the hop-permutation), as defined in 9.2.14.4. These subcarriers are henceforth referred to as Forward Dedicated Pilot Channel subcarriers. For the purpose of scrambling, the Forward Dedicated Pilot Channel subcarriers in each tile or quarter-tile shall be indexed by a quantity called the Forward Dedicated Pilot Channel index. The Forward Dedicated Pilot Channel index shall be computed according to the following procedure:

- a) Initialize an Orthogonal Frequency Division Multiplexing symbol counter i , a subcarrier counter j and a Forward Dedicated Pilot Channel index counter k to 0.
- b) If the subcarrier j in Orthogonal Frequency Division Multiplexing symbol i within the tile is a Forward Dedicated Pilot Channel subcarrier, then
 - 1) Set its Forward Dedicated Pilot Channel index to k .
 - 2) Increment k by 1.
- c) Increment i by 1. If $i = N_{\text{FRAME}}$, set i to 0 and increment j by 1.
- d) Repeat steps b) and c) until $j = N_{\text{BLOCK}}$.

In other words, the Forward Dedicated Pilot Channel subcarriers are indexed in time first, followed by frequency.

9.4.1.2.3.2.4.2 Scrambling sequence

The scrambling sequence for a tile depend on the tile index T which shall be equal to $(f_{\text{MIN}} - N_{\text{GUARD, LEFT}}) / N_{\text{BLOCK}}$, where f_{MIN} is the lowest indexed subcarrier in that tile. For the tile with index T , a complex scrambling sequence shall be generated using the common complex scrambling algorithm described in 9.2.5.2 with seed $f_{\text{PHY-HASH}}[38 \times 2^{20} \times 8 + \text{SectorSeed} \times 8 + T \bmod 8]$, where SectorSeed is as defined in 9.2.2.2.3. The k^{th} symbol $c(k)$ in the complex scrambling sequence shall be used to scramble the Forward Dedicated Pilot Channel subcarrier with Forward Dedicated Pilot Channel index k . The scrambling operation shall consist of multiplying the unscrambled complex symbol on the subcarrier with the scrambling symbol $c(k)$.

9.4.1.2.3.3 Forward Channel Quality Indicator Pilot Channel

The Forward Channel Quality Indicator Pilot Channel (F-CQIPICH) shall be present only in the BRCH zone when ResourceChannelMuxMode = 2. In this mode, the Forward Channel Quality Indicator Pilot Channel shall be present in Forward Link PHY frames satisfying $j \bmod \text{NumFLInterlaces} = 4$ when DuplexingMode = FDD, in the Forward Link PHY frames satisfying $j \bmod \text{NumFLInterlaces} = 0$ when DuplexingMode = TDD63 or TDD44. Here j denotes the index of the Forward Link PHY frame in the superframe. Additionally, if the DuplexingMode = FDD and FDDHalfDuplexTDDPartition is set to 1, then the Forward Channel Quality Indicator Pilot Channel shall also be present in the Forward Link PHY frames satisfying $j \bmod \text{NumFLInterlaces} = 3$. In these Forward Link PHY frames, the Forward Channel Quality Indicator Pilot Channel shall be present on the Orthogonal Frequency Division Multiplexing symbols with indices 3 and 4 in the Forward Link PHY frame, where the Orthogonal Frequency Division Multiplexing symbols in the Forward LinkPHY frame are indexed from 0 to 7. The Forward Channel Quality Indicator Pilot Channel is designed so as to enable the access terminal to estimate channel quality for reporting the r-cqich, and to estimate the optimal precoding matrix for reporting the r-bfch. The notion of a precoding matrix is defined in 9.2.8.2.

The Forward Channel Quality Indicator Pilot Channel is transmitted on a disjoint set of subcarriers from each effective antenna, with the number of effective antennas being given by the NumEffectiveAntennas field of the Overhead Messages Protocol. Each subcarrier occupied by the Forward Channel Quality Indicator Pilot Channel from a given effective antenna shall be modulated with the value $(\sqrt{P}, 0)$ from that effective antenna, where P is given by the CQIPilotTransmitPower field of the Overhead Messages Protocol. The remaining effective antennas shall be left unmodulated on this subcarrier.

For the Orthogonal Frequency Division Multiplexing symbol with index 3 within a Forward Link PHY frame containing the Forward Channel Quality Indicator Pilot Channel, a usable subcarrier with index i_{sc} shall be modulated with the Forward Channel Quality Indicator Pilot Channel from the antenna with index k if this subcarrier is not part of the Reserved Subzone or the DRCH zone, and if $i_{sc} \bmod 16 = a_k$. For the Orthogonal Frequency Division Multiplexing symbol with index 4 within a Forward Link PHY frame containing the Forward Channel Quality Indicator Pilot Channel, a usable subcarrier with index i_{sc} shall be modulated with the Forward Channel Quality Indicator Pilot Channel from the antenna with index k if this subcarrier is not assigned to a Reserved Subzone or the DRCH zone, and if $i_{sc} \bmod 16 = b_k$, where, a_k and b_k are as shown in Table 219, and have been chosen so as to ensure that the Forward Channel Quality Indicator Pilot Channel does not collide with the Forward Dedicated Pilot Channel. The Reserved Subzone is defined in 9.2.14.4.2.

Table 219—Values of the Parameters a_k and b_k

Antenna Index (k)	a_k	b_k
0	2	10
1	3	11
2	4	12
3	5	13
4	10	2
5	11	3
6	12	4
7	13	5

9.4.1.2.3.3.1 Forward Channel Quality Indicator Pilot Channel scrambling

For the purpose of scrambling, the Forward Channel Quality Indicator Pilot Channel subcarriers on each effective antenna shall be indexed by a quantity called the Forward Channel Quality Indicator Pilot Channel index. The Forward Channel Quality Indicator Pilot Channel index of a subcarrier on effective antenna k shall be computed according to the following procedure:

- a) Initialize an Orthogonal Frequency Division Multiplexing symbol counter i to 3, a subcarrier counter j and a Forward Channel Quality Indicator Pilot Channel index counter r to 0.
- b) If the subcarrier j in Orthogonal Frequency Division Multiplexing symbol i is a Forward Channel Quality Indicator Pilot Channel subcarrier, then
 - 1) Set its Forward Channel Quality Indicator Pilot Channel index on antenna k to r .
 - 2) Increment r by 1.
- c) Increment j by 1. If $j = N_{FFT}$, set j to 0 and increment i by 1.
- d) Repeat steps b) and c) until $i = 5$.

In other words, the Forward Channel Quality Indicator Pilot Channel subcarriers are indexed in frequency first, followed by time.

A complex scrambling sequence shall be generated using the common complex scrambling algorithm described in 9.2.5.2 with seed $f_{\text{PHY-HASH}}[2^{20} \times 8 \times 45 + \text{SectorSeed} \times 8 + k \bmod 8]$, where SectorSeed is as described in 9.2.2.2.3. The r^{th} symbol $c(r)$ in the complex scrambling sequence shall be used to scramble the Forward Channel Quality Indicator Pilot Channel subcarrier with the Forward Channel Quality Indicator Pilot Channel index r on effective antenna k . The scrambling operation shall consist of multiplying the unscrambled complex symbol on the subcarrier with the scrambling symbol $c(r)$.

9.4.1.2.3.4 Forward Cell Null Channel

The Forward Cell Null Channel (F-CNCH) defines subcarriers that are blanked by all the sectors in a cell. These subcarriers are used to measure the out-of-cell interference level.

In each OFDM symbol, the number of subcarriers $N_{\text{CNCH-SUBCARRIERS}}$ that shall be blanked in a given OFDM symbol shall be determined as follows:

- If the CellNullIDIncluded field of the Overhead Messages Protocol is equal to 0, then $N_{\text{CNCH-SUBCARRIERS}} = 0$.
- Otherwise $N_{\text{CNCH-SUBCARRIERS}} = 2$

The subcarriers to be blanked in PHY frame j within superframe i shall be chosen according to the following procedure:

- a) Set $\text{SEED}_{\text{CNCH}} = f_{\text{PHY-HASH}}(33 \times 64 \times 1024 \times 16 + (j \bmod 64) \times 1024 \times 16 + (i \bmod 16) \times 1024 + \text{CellNullID})$, where $f_{\text{PHY-HASH}}$ is the common hash function described in 9.2.5.4 and CellNullID is a field of the Overhead Messages Protocol.
- b) Generate a permutation H_{CNCH} of size $(N_{\text{FFT}} - N_{\text{GUARD}})$ using the common permutation generation algorithm (9.2.5.1) with seed $\text{SEED}_{\text{CNCH}}$.
- c) Initialize an OFDM symbol counter t to 0. Initialize an intra-symbol subcarrier counter k_{INTRA} to 0, and an overall subcarrier counter k_{OVERALL} to 0.
- d) If subcarrier $H_{\text{CNCH}}(k_{\text{OVERALL}})$ in OFDM symbol t within PHY frame j in superframe i is not allocated to the Forward Dedicated Pilot Channel or the Forward Channel Quality Indicator Pilot Channel, then
 - 1) Allocate this subcarrier to the Forward Cell Null Channel.
 - 2) Increment k_{INTRA} by 1.
 - 3) If k_{INTRA} is equal to $N_{\text{CNCH-SUBCARRIERS}}$, then set k_{INTRA} to 0 and increment t by 1.
 - 4) If $t = N_{\text{FRAME}}$, stop.
- e) Increment k_{OVERALL} by 1 and go to step d).

The access network shall puncture all subcarriers allocated to the Forward Cell Null Channel, i.e., no energy shall be transmitted on these subcarriers even if these subcarriers would have been modulated by other channels.

9.4.1.2.3.5 Forward Beacon Pilot Channel

The Forward Beacon Pilot Channel (F-BPICH) is used to indicate the presence of the access network to access terminals on other ChannelBands. A sector may transmit the Forward Beacon Pilot Channel on one or more ChannelBands, where ChannelBand is obtained from a mapping of ShortChannelIDs to ChannelBands. ShortChannelID is a field of the Overhead Messages Protocol.

The modulation of the Forward Beacon Pilot Channel depends on the following parameters:

- $N_{\text{CHANNELBANDS}}$, which is equal to the NumShortChannelIDs field of the Overhead Messages Protocol.
- $I_{\text{CHANNELBAND}}$, which is equal to the ShortChannelID corresponding to the ChannelBand on which the Forward Beacon Pilot Channel is being transmitted. The Forward Beacon Pilot Channel shall not be transmitted on a ChannelBand that does not have an assigned ShortChannelID.
- $[r_1r_0]$, which is equal to the ShortChannelID corresponding to the preferred ChannelBand of the sector. The preferred ChannelBand is the ChannelBand to which an access terminal should tune to in order to demodulate the Overhead Messages of that sector.

On a given ChannelBand, say ChannelBand X, some sectors may transmit only the Forward Beacon Pilot Channel. Such sectors are referred to as a BeaconOnlySector. Typically, a BeaconOnlySector transmits its Overhead Messages (and other channels) on another ChannelBand, say ChannelBand Y. In such a case, the Forward Beacon Pilot Channel on ChannelBand X is used to indicate to the access terminals operating on ChannelBand X that this sector is present on ChannelBand Y. This helps the access terminals in monitoring the sectors on ChannelBand Y without the need for tuning away to that ChannelBand i.e., while they are still operating on ChannelBand X. In the case of BeaconOnlySectors, the modulation of Forward Beacon Pilot Channel still depends on the parameters $N_{\text{CHANNELBANDS}}$, $I_{\text{CHANNELBAND}}$ and $[r_1r_0]$. If an access terminal detects the transmission of the Forward Beacon Pilot Channel by a new sector and decodes the above parameters, then it should use the ChannelList to infer which ChannelBand is preferred by the new sector.

9.4.1.2.3.5.1 Forward Beacon Pilot Channel encoding

Beacon Code A shall be used to encode the Forward Beacon Pilot Channel when $(N_{\text{FFT}} - N_{\text{GUARD}}) \geq 422$ and Beacon Code B shall be used when $(N_{\text{FFT}} - N_{\text{GUARD}}) < 422$.

9.4.1.2.3.5.1.1 Beacon Code A

Beacon Code A is a Reed-Solomon code that encodes a 12-bit quantity M into a sequence of non-binary numbers in the set $\{0, 1, 2, \dots, Q-1\}$, where $Q = 211$. The t^{th} number in the sequence shall be given by

$$X_t(M) = (p_1^{\alpha_1 + 21t} + p_1^{\alpha_2 + 42t}) \bmod Q, \text{ where } p_1 = 207, \alpha_1 = \lfloor M/(Q-1) \rfloor \text{ and } \alpha_2 = M \bmod (Q-1).$$

Note that p_1 is a primitive element of $\text{GF}(Q)$. Therefore $(p_1^{Q-1} \bmod Q) = 1$.

9.4.1.2.3.5.1.2 Beacon Code B

Beacon Code B is a Reed-Solomon code that encodes a 12-bit quantity M into a sequence of non-binary numbers in the set $\{0, 1, 2, \dots, Q-1\}$, where $Q = 47$. The t^{th} number in the sequence shall be given by

$$X_t(M) = (p_1^{\alpha_1 + 2t} + p_1^{\alpha_2 + 4t} + p_1^{\alpha_3 + 6t}) \bmod Q, \text{ where } p_1 = 45, \alpha_1 = \lfloor M/(Q-1)^2 \rfloor \text{ and } \alpha_2 = \lfloor M/(Q-1) \rfloor \bmod (Q-1) \text{ and } \alpha_3 = M \bmod (Q-1).$$

Note that p_1 is a primitive element of $\text{GF}(Q)$. Therefore $(p_1^{Q-1} \bmod Q) = 1$.

9.4.1.2.3.5.2 Forward Beacon Pilot Channel transmission

The generation of BeaconOnlyOFDMSymbols is performed differently for the two values of the FDDHalfDuplexTDDPartition field of the Overhead Messages Protocol.

9.4.1.2.3.5.2.1 BeaconOnlyOFDMSymbols for all cases except for DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 1

If DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 1, and the SinglePAForMultipleChannelBands field of the Overhead Messages Protocol is set to 1, then there shall be $N_{\text{CHANNELBANDS}}$ OFDM symbols every two superframes. These symbols are referred to as “BeaconOnlyOFDMSymbols.” The BeaconOnlyOFDMSymbols are indexed from 0 through $N_{\text{CHANNELBANDS}} - 1$, and shall be chosen according to the following procedure:

- Set $\text{SEED}_{\text{BEACON-FRAMES}} = f_{\text{PHY-HASH}}(78 \times 512 + \text{PilotPN})$. Let $H_{\text{BEACON-FRAMES}}$ be the permutation of size $2N_{\text{PHYFrames}}$ generated using the common permutation generation algorithm (9.2.5.1) with seed $\text{SEED}_{\text{BEACON-FRAMES}}$.
- Initialize a counter i and a counter j to 0.
- Set $\text{SEED}_{\text{BEACON-SYMBOLS}} = f_{\text{PHY-HASH}}(79 \times 512 \times 8 + (i \bmod 8) \times 512 + \text{PilotPN})$. Let $\text{TMP} = \text{SEED}_{\text{BEACON-SYMBOLS}} \bmod 2$.
- Initialize a counter j to 0. If the PHY frame $H_{\text{BEACON-FRAMES}}(j) \bmod N_{\text{PHYFrames}}$ contains the Forward Channel Quality Indicator Pilot Channel, increment j by 1.
- If $H_{\text{BEACON-FRAMES}}(j) < N_{\text{PHYFrames}}$, the OFDM symbol $(3 + \text{TMP})^{29}$ in PHY frame $H_{\text{BEACON-FRAMES}}(i) \bmod N_{\text{FLPHYFrames}}$ in all even indexed superframes shall be the BeaconOnlyOFDMSymbol with index i . If $H_{\text{BEACON-FRAMES}}(i) \geq N_{\text{PHYFrames}}$, the OFDM symbol $(3 + \text{TMP})$ in PHY frame $H_{\text{BEACON-FRAMES}}(i) \bmod N_{\text{PHYFrames}}$ in all odd indexed superframes is referred to as the beacon OFDM symbol with index i .
- Increment i by 1. If $i = N_{\text{CHANNELBANDS}}$, stop. Otherwise go to step c).

If the SinglePAForMultipleChannelBands field of the Overhead Message Parameter is set to 0, there is only one BeaconOnlyOFDMSymbol every two superframes. This symbol shall be equal to the BeaconOnlyOFDMSymbol with index $I_{\text{CHANNELBAND}}$ as computed in the above procedure.

No channels other than the Forward Beacon Pilot Channel shall be transmitted on BeaconOnlyOFDMSymbols.

9.4.1.2.3.5.2.2 BeaconOnlyOFDMSymbols for FDDHalfDuplexTDDPartition = 1

If FDDHalfDuplexTDDPartition is set to 1 and the SinglePAForMultipleChannelBands field of the Overhead Messages Protocol is set to 1, then there shall be $2N_{\text{CHANNELBANDS}}$ BeaconOnlyOFDMSymbols every two superframes.

If FDDHalfDuplexTDDPartition is set to 1, there shall be $2N_{\text{CHANNELBANDS}}$ BeaconOnlyOFDMSymbols if SinglePAForMultipleChannelBands field of the Overhead Messages Protocol is set to 1. The BeaconOnlyOFDMSymbols with indices 0 through $N_{\text{CHANNELBANDS}} - 1$ shall be identical to the BeaconOnlyOFDMSymbols generated as described in 9.4.1.2.3.5.2.1. The BeaconOnlyOFDMSymbol with index $N_{\text{CHANNELBANDS}} + i$, where $0 \leq i < N_{\text{CHANNELBANDS}} - 1$ shall be generated as follows:

- Let the OFDM symbol indexed t in the PHY frame j within a superframe be the BeaconOnlyOFDMSymbol with index i .
- Initialize a counter $n = N_{\text{CHANNELBANDS}} + i$.
- Let m be the Reverse Link PHY frame with index $H_{\text{BEACON-FRAMES}}[n] \bmod N_{\text{PHYFrames}}$ in even indexed superframes if $H_{\text{BEACON-FRAMES}}[n] < N_{\text{PHYFrames}}$ and in odd indexed superframes otherwise. Increment n by 1 and repeat step c) if one or more of the following conditions are satisfied.
 - PHY frame m contains the Forward Channel quality Indicator Pilot Channel.

²⁹Note that F-DPICH pilot formats 0 and 1 are not present on OFDM symbols 3 and 4.

- 2) PHY frame m is on the same half-duplex interlace as PHY frame j .
- 3) PHY frame m contains one of the BeaconOnlyOFDMSymbols with indices 0 through $N_{\text{CHANNELBANDS}} + i - 1$.
- d) The OFDM symbol with index t in the PHY frame $(j+1) \bmod NN_{\text{PHYFrames}}$ is referred to as the BeaconOnlyOFDMSymbol with index $N_{\text{CHANNELBANDS}} + i$.

If the SinglePAForMultipleChannelBands field of the Overhead Message Parameter is set to 0, there shall only be two BeaconOnlyOFDMSymbol every 2 superframes. These symbols shall be equal to the BeaconOnlyOFDMSymbols with index $I_{\text{CHANNELBAND}}$ and $N_{\text{CHANNELBANDS}} + I_{\text{CHANNELBAND}}$ as computed in the above procedure.

9.4.1.2.3.5.2.3 Beacon Subcarrier Groups

The set of usable subcarriers in a BeaconOnlyOFDMSymbol is divided into $N_{\text{BEACON\_SUBCARRIER\_GROUPS}}$ groups, where

$$N_{\text{BEACON\_SUBCARRIER\_GROUPS}} = \left\lfloor \frac{N_{\text{FFT}} - N_{\text{GUARD}}}{Q} \right\rfloor,$$

where $Q = 211$ for beacon code A and $Q = 47$ for beacon code B. The beacon subcarrier groups shall be indexed 0 through $N_{\text{BEACON\_SUBCARRIER\_GROUPS}} - 1$. The beacon subcarrier group with index k shall comprise of subcarriers $f_{\text{START}}(k)$ through $f_{\text{START}}(k) + Q - 1$ where

$$f_{\text{START}}(k) = \frac{N_{\text{FFT}}}{2} - \left\lfloor \frac{N_{\text{BEACON\_SUBCARRIER\_GROUPS}} Q}{2} \right\rfloor + kQ$$

The subcarriers within group k shall be indexed 0 through $Q - 1$.

9.4.1.2.3.5.2.4 Forward Beacon Pilot Channel modulation

The Forward Beacon Pilot Channel shall be modulated according to the following procedure:

- a) A value of $\sqrt{P}$ shall be transmitted on the subcarrier with index $X_t(M)$ within the beacon subcarrier group with index k on the BeaconOnlyOFDMSymbol with index $I_{\text{CHANNELBAND}}$.
 - 1) The modulation may occur on any linear combination of effective antennas. The choice of linear combination is beyond the scope of this standard.
 - 2) The values of P shall be greater than N_{FFT} . The exact value of P is beyond the scope of this standard.
 - 3) The value of k is beyond the scope of this standard.
 - 4) In all cases except for DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 1, then $M = [r_1 r_0 b_9 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0]$, where $[b_9 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0]$ is the 10-bit PilotID and $[r_1 r_0]$ is the Preferred ShortChannelID as described in 9.4.1.2.3.5.
 - 5) If DuplexingMode = FDD and FDDHalfDuplexTDDPartition = 1, $M = [0 r_1 r_0 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0]$ where $[b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0]$ is the 9-bit PilotPN and $[r_1 r_0]$ is the Preferred ShortChannelID as described in 9.4.1.2.3.5]
 - 6) Define $t = \lfloor \text{superframe index} / 2 \rfloor$ and $X_t(M)$ is generated as described in 9.4.1.2.3.5.
- b) If FDDHalfDuplexTDDPartition is set to 1, a value of $\sqrt{P}$ shall be transmitted on the subcarrier with index $X_t'(M')$ within the beacon subcarrier group with index k' on the BeaconOnlyOFDMSymbol with index $(N_{\text{CHANNELBANDS}} + I_{\text{CHANNELBAND}})$.
 - 1) The value of k' is beyond the scope of this standard.

- 2) $M' = [1r_1r_0b_8b_7b_6b_5b_4b_3b_2b_1b_0]$, where $[b_8b_7b_6b_5b_4b_3b_2b_1b_0]$ is the 9-bit PilotPN and $[r_1r_0]$ is the preferred ShortChannelID described in 9.4.1.2.3.5.
- 3) Define $t' = \lfloor \text{superframe index}/2 \rfloor + 5$ if Beacon Code A is used and $t' = \lfloor \text{superframe index}/2 \rfloor + 11$ if Beacon Code B is used, where the generation of $X_t'(M')$ is described in 9.4.1.2.3.5.
- c) No power shall be transmitted on all other subcarriers.

9.4.1.2.4 Forward Link Control Channels in the PHY frames

In every PHY frame in a superframe, the Forward Link control channels, i.e., in the Forward Acknowledgment Channel, the Forward Start-of-Packet Channel, the Forward Reverse Activity Bit Channel, the Forward Shared Control Channel, the Forward Fast Other Sector Interference Channel, the Forward Interference over Thermal Channel, the Forward Pilot Quality Indicator Channel and the Forward Power Control Channel shall be multiplexed together onto a set of $N_{\text{FLCS-BLOCKS}}$ hop-port blocks, referred to as the “Forward Link Control Segment.”

The value of $N_{\text{FLCS-BLOCKS}}$ is given by $(N_{\text{FLCS-COMMON-BLOCKS}} + 3 \times N_{\text{FLCS-LAB-SEGMENTS}})$, where $N_{\text{FLCS-COMMON-BLOCKS}}$ and $N_{\text{FLCS-LAB-SEGMENTS}}$ are given by the NumCommonSegmentHopPortBlocks and NumLABSegments fields of the Overhead Messages Protocol respectively. The hop-port blocks shall be indexed 0 though $N_{\text{FLCS-BLOCKS}} - 1$, and the hop-ports in these blocks shall be referred to as “Forward Link Control Segment hop-ports”. The Forward Link Control Segment blocks shall either all be BRCH resources or all be DRCH resources. The choice of BRCH or DRCH is specified by the UseDRCHForFLCS field of the Overhead Messages Protocol.

9.4.1.2.4.1 Forward Link Control Segment available subcarriers

Not all subcarriers may be available for modulation by the Forward Link Control Segment. For example, subcarriers in which the Forward Dedicated Pilot Channel is transmitted cannot be used by the Forward Link Control Segment. In this subclause, the notion of Forward Link Control Segment unavailable subcarriers is defined. A subcarrier is unavailable for the Forward Link Control Segment if:

- The subcarrier is a pilot subcarrier i.e., it is allocated to one of the Forward Link Pilot Channels (Forward Dedicated Pilot Channel, Forward Channel Quality Indicator Pilot Channel, and Forward Common Pilot Channel)
- The subcarrier is part of a BeaconOnlyOFDMSymbol.

Furthermore, if ResourceChannelMuxMode = 1, all subcarriers allocated to the DRCH are not available when modulating BRCH resources.

All subcarriers that are not unavailable as defined above shall be referred to as “Forward Link Control Segment available subcarriers.”

9.4.1.2.4.2 Forward Acknowledgment Channel

The Forward Acknowledgment Channel (F-ACKCH) is primarily used to acknowledge Reverse Link HARQ transmissions and is present in every Forward Link PHY frame to acknowledge the associated Reverse Link PHY frame. Each value transmitted on the Forward Acknowledgment Channel in a given Forward Link PHY frame is associated with a unique FACKNodeIndex, which is determined by the FLCS MAC Protocol (see 8.5). For every Forward Link PHY frame, the FLCS MAC Protocol (see 8.5) also determines the total number of acknowledgements to be carried in that Forward Link PHY frame, which is denoted by TotalFACKNodeIndices. For each value of FACKNodeIndex, the FLCS MAC Protocol (see 8.5) also determines the quantities FACKVal, P_{ACKCH} , and MACID. FACKVal denotes the acknowledgement value and takes values 0, 1, 2, and 3, P_{ACKCH} denotes the power density at which the Forward Acknowledgment Channel is to be transmitted on the FACKNodeIndex of interest, and MACID

denotes the MACID of the terminal to which the Forward Acknowledgement Channel on that FACKNodeIndex is targeted.

The Forward Acknowledgement Channel shall be transmitted on resources 0 through $N_{\text{FACKH-INDICES}} - 1$, where $N_{\text{FACKH-INDICES}} = \lceil 4 \times \text{TotalFACKNodeIndices}/4 \rceil$.

9.4.1.2.4.2.1 Forward Acknowledgment Channel transmission

A sequence of length 12 shall be transmitted on the Forward Acknowledgment Channel for each value of FACKNodeIndex received from the FLCS MAC Protocol (see 8.5). The FLCS resources used for the transmission of this sequence depend on the value of FACKNodeIndices. If the sequences corresponding to different values of FACKNodeIndex use the same transmission resources, then those resources shall be superimposed on these resources.

The length 12 sequence shall be denoted $\{Z_{00}, Z_{01}, Z_{02}, Z_{10}, Z_{11}, Z_{12}, Z_{20}, Z_{21}, Z_{22}, Z_{30}, Z_{31}, Z_{32}\}$ and shall be constructed as follows:

- a) Let $\alpha = 0$ if FACKVal is 0, and $\alpha = \exp((2\pi j/3) \times \text{FACKVal})$ otherwise.
- b) Define $\text{SEED}_{\text{FACKCH}} = f_{\text{PHY-HASH}}(2^{20} \times 2048 \times 3 + 2048 \times \text{SectorSeed} + \text{MAC\_ID})$, where SectorSeed is as defined in 9.2.2.2.3 and MAC_ID is the MACID of the access terminal of interest.
- c) Let $[Y_0 \ Y_1 \ Y_2]$ be the sequence of length 3 generated using the common complex scrambling algorithm in 9.2.5.2 using seed $\text{SEED}_{\text{FACKCH}}$.
- d) Let $r = \text{FACKNodeIndex} \bmod 4$ and $R_{\text{FACK-TRANS}} = 4 \times \lfloor \text{FACKNodeIndex}/4 \rfloor$.
- e) If the Forward Link Control Segment resource with index $R_{\text{FACK-TRANS}}$ is a BRCH resource, then let $[D_0 \ D_1 \ D_2 \ D_3]$ be the column with index r of the DFT matrix of size 4 as defined in 9.2.5.5. If the Forward Link Control Segment resource with index $R_{\text{FACK-TRANS}}$ is a DRCH resource, then let $[D_0 \ D_1 \ D_2 \ D_3]$ be the column with index r in the 4×4 identity matrix.
- f) $Z_{ij} = \alpha \sqrt{P_{\text{ACKCH}}} \times D_i Y_j$ for $0 \leq i < 4$ and $0 \leq j < 3$.
- g) Z_{ij} shall be used to populate subcarrier symbol j in the Forward Link Control Segment resource ($i + R_{\text{FACK-TRANS}}$), provided that subcarrier-symbol is mapped to a Forward Link Control Segment available subcarrier.

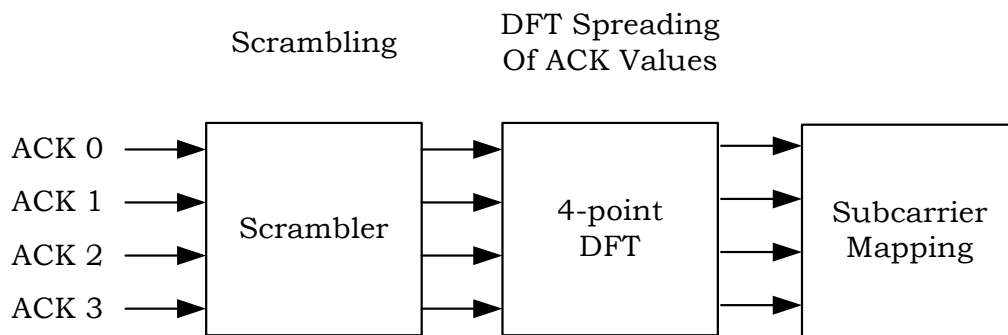


Figure 136—ACK Processing for FACKNodeIndices 0 through 3

9.4.1.2.4.3 Forward Start-of-Packet Channel

The Forward Start-of-Packet Channel (F-SPCH) is used by the access network to send start-of-packet indications to an access terminal. A start-of-packet indication is used to retain a Forward Link persistent assignment of an access terminal in the absence of active data transmission, or to signal the start of a new packet on the persistent assignment, or to deassign the persistent assignment.

Each value transmitted on the Forward Start-of-Packet Channel in a given Forward Link PHY frame is associated with a unique FSPNodeIndex, which is determined by the FLCS MAC Protocol (see 8.5). For every Forward Link PHY frame, the FLCS MAC Protocol (see 8.5) also determines the total number of values to be carried in that Forward Link PHY frame, which is denoted by TotalFSPNodeIndices. For each value of FSPNodeIndex, the FLCS MAC Protocol (8.5) also determines the quantities FSPVal, P_{FSPCH} , and MACID. FSPVal denotes the value to be transmitted and takes values 0, 1, 2 and 3, P_{FSPCH} denotes the power density at which the Forward Start-of-Packet Channel is to be transmitted on the FSPNodeIndex of interest.

The Forward Start-of-Packet Channel shall be transmitted on resources $R_{\text{FSPCH-BEGIN}}$ through $R_{\text{FSPCH-BEGIN}} + N_{\text{FSPCH-INDICES}} - 1$, where $R_{\text{FSPCH-BEGIN}} = N_{\text{FACK-INDICES}}$ and $N_{\text{FSPCH-INDICES}} = 4 \times \lfloor \text{TotalFSPNodeIndices} \rfloor$.

9.4.1.2.4.3.1 Forward Start-of-Packet Channel transmission

A sequence of length 12 shall be transmitted on the Forward Start-of-Packet Channel for each value of FSPNodeIndex received from the FLCS MAC Protocol (8.5). The FLCS resources used for the transmission of this sequence depend on the value of FSPNodeIndex. If the sequences corresponding to different values of FSPNodeIndex use the same transmission resources, then these sequences shall be superimposed on these resources.

The length 12 sequence is denoted by $\{Z_{00}, Z_{01}, Z_{02}, Z_{10}, Z_{11}, Z_{12}, Z_{20}, Z_{21}, Z_{22}, Z_{30}, Z_{31}, Z_{32}\}$ and shall be constructed as follows:

- Let $\alpha = 0$ if FSPVal is 0, and $\alpha = \exp((2\pi j/3) \times \text{FSPVal})$ otherwise.
- Define $\text{SEED}_{\text{FSPCH}} = f_{\text{PHY-HASH}}(2^{20} \times 2 + \text{SectorSeed})$, where SectorSeed is as defined in 9.2.2.2.3.
- Let $[Y_0 \ Y_1 \ Y_2]$ be the sequence of length 3 generated using the common complex scrambling algorithm in 9.2.5.2 using seed $\text{SEED}_{\text{FSPCH}}$.
- Let $r = \text{FSPNodeIndex} \bmod 4$ and $R_{\text{FSP-TRANS}} = R_{\text{FSPCH-BEGIN}} + 4 \times \lfloor \text{FSPNodeIndex}/4 \rfloor$.
- If the Forward Link Control Segment resource with index $R_{\text{FSP-TRANS}}$ is a BRCH resource, then let $[D_0 \ D_1 \ D_2 \ D_3]$ be the column with index r of the DFT matrix of size 4 as defined in 9.2.5.5. If the Forward Link Control Segment resource with index $R_{\text{FSP-TRANS}}$ is a DRCH resource, then let $[D_0 \ D_1 \ D_2 \ D_3]$ be the column with index r in the 4x4 identity matrix.
- $Z_{ij} = \alpha \sqrt{P_{\text{ACKCH}}} \times D_i Y_j$ for $0 \leq i < 4$ and $0 \leq j < 3$.

Z_{ij} shall be used to populate subcarrier symbol j in the Forward Link Control Segment resource ($R_{\text{FSP-TRANS}} + i$), provided that subcarrier-symbol is mapped to a Forward Link Control Segment available subcarrier.

9.4.1.2.4.4 Forward Reverse Activity Bit Channel

The Forward Reverse Activity Bit Channel (F-RABCH) carries an one bit indication about the load on the Reverse Code Division Multiple Access segment. This aids the access terminal to choose when to transmit on the Reverse Code Division Multiple Access segment.

The FLCS MAC Protocol (see 8.5) determines the number of reports, denoted by TotalFRABReports, to be sent on the Forward Reverse Activity Bit Channel in each Forward Link PHY frame. For each report, the FLCS MAC Protocol (see 8.5) also provides an index FRABReportIndex, a one-bit value RABVal, and the power density P_{FRABCH} at which this value shall be transmitted. The Forward Reverse Activity Bit Channel shall be transmitted on resources $R_{\text{FRABCH-BEGIN}}$ through $R_{\text{FRABCH-BEGIN}} + 2 \times N_{\text{FRABCH-INDICES}} - 1$, where $R_{\text{FRABCH-BEGIN}} = R_{\text{FSPCH-BEGIN}} + N_{\text{FSPCH-INDICES}}$ and $N_{\text{FRABCH-INDICES}}$ is equal to TotalFRABReports.

9.4.1.2.4.4.1 Forward Reverse Activity Bit channel modulation

The RABVal received from the FLCS MAC Protocol (8.5) shall be mapped to a modulation symbol b using BPSK modulation. A sequence of six modulation symbols $\{s_0, s_1, s_2, s_3, s_4, s_5\}$ is generated using the complex scrambling algorithm described in 9.2.5.3 with input seed $f_{\text{PHY-HASH}}(2^{20} \times 49 + \text{SectorSeed})$, where SectorSeed is as defined in 9.2.2.2.3. Define the six symbol sequence $\{c_0, c_1, c_2, c_3, c_4, c_5\}$, $c_i = s_i b$, $i = 0, 1, \dots, 5$.

9.4.1.2.4.4.2 Forward Reverse Activity Bit Channel resource allocation

For $i = 0, 1, \dots, 5$; a value of $\sqrt{P_{\text{FRABCH}}} \times c_i$ shall be transmitted on subcarrier-symbol $\lfloor i/2 \rfloor$ in the Forward Link Control Segment resource with index $R_{\text{FRABCH-BEGIN}} + 2 \times i \pmod{2}$, provided that hop-port is mapped to a Forward Link Control Segment available subcarrier, where i denotes the FRABReportIndex.

9.4.1.2.4.5 Forward Pilot Quality Indicator Channel

The Forward Pilot Quality Indicator Channel (F-PQICH) carries quantized values of the Reverse Link pilot strength for each access terminal. This aids the access terminal to select the optimal serving sector, and is used to control the power level of the Reverse Link control and data channels.

The FLCS MAC Protocol (see 8.5) determines the number of Forward Pilot Quality Indicator Channel reports to be transmitted on each Forward Link PHY frame, which is denoted by TotalFPQIReports. Each of these blocks is assigned an index FPQIReportIndex that takes values 0 through TotalFPQIReports – 1. For each FPQIReportIndex, the FLCS MAC Protocol (see 8.5) determines a four-bit value FPQIVal to be transmitted, the ID of the terminal to which this report is targeted and the power density P_{FPQICH} at which it is sent.

The Forward Pilot Quality Indicator Channel shall be transmitted on resources $R_{\text{FPQICH-BEGIN}}$ through $R_{\text{FPQICH-BEGIN}} + 2N_{\text{FPQICH-INDICES}} - 1$, where $R_{\text{FPQICH-BEGIN}} = R_{\text{FRABCH-BEGIN}} + 2N_{\text{FRABCH-INDICES}}$ and $N_{\text{FPQICH-INDICES}}$ is given by TotalFPQIReports.

9.4.1.2.4.5.1 Forward Pilot Quality Indicator Channel Encoding

The four bit value FPQIVal shall be encoded by a rate-1/3 concatenated code whose codewords are as specified in 9.2.6.3.2.

9.4.1.2.4.5.2 Forward Pilot Quality Indicator Channel Modulation

The encoded data shall be QPSK modulated as specified in 9.2.6.7.1 to yield a sequence of symbols $\{b_0, b_1, b_2, b_3, b_4, b_5\}$. A sequence $\{s_0, s_1, s_2, s_3, s_4, s_5\}$ is generated using the complex scrambling algorithm described in 9.2.5.3 with input seed $f_{\text{PHY-HASH}}(2^{31} \times 47 + 2^{20} \times \text{MACID} + \text{SectorSeed})$, where MACID is the MAC identifier of the access terminal, and SectorSeed is defined in 9.2.2.2.3. Define the 6 symbol sequence $\{c_0, c_1, c_2, c_3, c_4, c_5\}$ where $c_i = s_i b_i$, $i = 0, 1, \dots, 5$.

9.4.1.2.4.5.3 Forward Pilot Quality Indicator Channel Resource Allocation

For $i = 0, 1, \dots, 5$; a value of $\sqrt{P_{\text{FPQICH}}} \times c_i$ shall be transmitted on subcarrier-symbol $\lfloor i/2 \rfloor$ in the Forward Link Control Segment resource with index $R_{\text{FPQICH-BEGIN}} + 2 \times \text{FPQIMessageIndex} + (i \pmod{2})$, provided that the subcarrier-symbol is mapped to a Forward Link Control Segment available subcarrier with $P = P_{\text{FPQICH}}$.

9.4.1.2.4.6 Forward Fast Other-Sector-Interference Channel

The Forward Fast Other Sector Interference Channel (F-FOSICH) is used to indicate interference levels in a given Reverse Link hop-port subzone to the access terminals in other sectors.

The FLCS MAC Protocol (see 8.5) determines the number of the Forward Fast Other Sector Interference Channel reports to be transmitted on each Forward Link PHY frame, which is denoted by TotalFFOSIReports. Each of these blocks is assigned an index FFOSIReportIndex that takes values 0 through TotalFFOSIReports – 1. For each FFOSIReportIndex, the FLCS MAC Protocol (8.5) determines a four-bit value FFOSIVal to be transmitted and the power density P_{FOSICH} at which it is sent.

The Forward Fast Other Sector Interference Channel shall be transmitted on resources $R_{\text{FFOSICH-BEGIN}}$ through $R_{\text{FFOSICH-BEGIN}} + 2N_{\text{FFOSICH-INDICES}} - 1$, where $R_{\text{FFOSICH-BEGIN}} = R_{\text{FPQICH-BEGIN}} + 2N_{\text{FPQICH-MESSAGES}}$ and $N_{\text{FFOSICH-INDICES}} = \text{TotalFFOSIReports}$.

9.4.1.2.4.6.1 Forward Fast Other-Sector-Interference Channel encoding

If the FFOSIVal is not equal to 0000, FFOSIVal shall be encoded by a rate-1/3 concatenated code whose codewords are as specified in 9.2.6.3.2.

9.4.1.2.4.6.2 Forward Fast Other-Sector-Interference Channel modulation

The encoded data shall be QPSK modulated as specified in 9.2.6.7.1 to yield a sequence of symbols $\{b_0, b_1, b_2, b_3, b_4, b_5\}$. A sequence $\{s_0, s_1, \dots, s_5\}$ is generated using the complex scrambling algorithm described in 9.2.5.2 with input seed $f_{\text{PHY-HASH}}(2^{20} \times 47 + \text{SectorSeed})$, where SectorSeed is defined in 9.2.2.2.3. Define the 6 symbol sequence $\{c_0, c_1, c_2, c_3, c_4, c_5\}$ where $c_i = s_i b_i$, where $i = 0, 1, \dots, 5$.

If the FFOSIMessage is equal to '0000', define $\{c_0, c_1, c_2, c_3, c_4, c_5\}$ to be $\{0, 0, 0, 0, 0, 0\}$.

9.4.1.2.4.6.3 Forward Fast Other-Sector-Interference Channel resource allocation

For $i = 0, 1, \dots, 5$; a value of $\sqrt{P_{\text{FOSICH}}} \times c_i$ shall be transmitted on subcarrier-symbol $\lfloor i/2 \rfloor$ in the Forward Link Control Segment resource with index $(R_{\text{FFOSICH-BEGIN}} + 2 \times \text{FFOSIReportIndex} + (i \bmod 2))$, provided that subcarrier-symbol is mapped to a Forward Link Control Segment available subcarrier.

9.4.1.2.4.7 Forward Interference-Over-Thermal Channel

The Forward Interference over Thermal Channel (F-IOTCH) is used to indicate interference levels in a given subzone to access terminals.

The FLCS MAC Protocol (see 8.5) determines the number of Forward Interference-Over-Thermal Channel reports to be transmitted on each Forward Link PHY frame, which is denoted by TotalFIOTReports. Each of these blocks is assigned an index FIOTReportIndex that takes values 0 through TotalFIOTReports – 1. For each FIOTReportIndex, the FLCS MAC Protocol (see 8.5) determines a four-bit value IOTVal to be transmitted and the power density P_{FIOTCH} at which it is sent.

The Forward Interference-Over-Thermal Channel shall be transmitted on resources $R_{\text{FIOTCH-BEGIN}}$ through $R_{\text{FIOTCH-BEGIN}} + 2N_{\text{FIOTCH-INDICES}} - 1$, where $N_{\text{FIOTCH-BEGIN}} = R_{\text{FFOSICH-BEGIN}} + 2N_{\text{FFOSICH-INDICES}}$ and $N_{\text{FIOTCH-INDICES}} = \text{TotalFIOTReports}$.

9.4.1.2.4.7.1 Forward Interference-Over-Thermal Channel encoding

The Forward Interference-over-Thermal Channel data, IOTVal, shall be encoded by a rate-1/3 concatenated code whose codewords are as specified in 9.2.6.3.2.

9.4.1.2.4.7.2 Forward Interference-Over-Thermal Channel modulation

The encoded data shall be QPSK modulated as specified in 9.2.6.7.1 to yield a sequence of symbols $\{b_0, b_1, b_2, b_3, b_4, b_5\}$. A sequence $\{s_0, s_1, \dots, s_5\}$ is generated using the complex scrambling algorithm described in 9.2.5.2 with input seed $f_{\text{PHY-HASH}}(2^{20} \times 43 + \text{SectorSeed})$, where SectorSeed is defined in 9.2.2.2.3. Define the 6 symbol sequence $\{c_0, c_1, c_2, c_3, c_4, c_5\}$ where $c_i = s_i b_i$, where $i = 0, 1, \dots, 5$.

9.4.1.2.4.7.3 Forward Interference-over-Thermal Channel resource allocation

For $i = 0, 1, \dots, 5$, a value of $\sqrt{P_{\text{FIOTCH}}} \times c_i$ shall be transmitted on subcarrier-symbol $\lfloor i/2 \rfloor$ in the Forward Link Control Segment resource with index $(R_{\text{FIOTCH-BEGIN}} + 2 \times \text{FIOTReportIndex} + (i \bmod 2))$, provided that subcarrier-symbol is mapped to a Forward Link Control Segment available subcarrier.

9.4.1.2.4.8 Forward Power Control Channel

The Forward Power Control Channel (F-PCCH) carries commands for closed-loop control of the Reverse Link control channel transmit power.

The FLCS MAC Protocol (see 8.5) determines the number of the Forward Power Control Channel reports to be transmitted on each Forward Link PHY frame, which is denoted by TotalFPCReports. Each of these blocks is assigned an index FPCReportIndex, which takes values 0 through TotalFPCReports – 1. For each FPCReportIndex, the FLCS MAC Protocol (see 8.5) determines a one-bit value PCVal to be transmitted, the MACID of the terminal to which this report is targeted and the power density P_{FPCCH} at which it is sent.

The Forward Power Control Channel shall be transmitted on the Forward Link Control Segment resources $R_{\text{FPCCH-BEGIN}}$ through $R_{\text{FPCCH-BEGIN}} + \lceil N_{\text{FPCCH-INDICES}} / 3 \rceil - 1$, where $R_{\text{FPCCH-BEGIN}} = R_{\text{FIOTCH-BEGIN}} + 2N_{\text{FIOTCH-INDICES}}$ and $N_{\text{FPCCH-INDICES}}$ is given by TotalFPCReports.

9.4.1.2.4.8.1 Forward Power Control Channel transmission

The first symbol of the scrambling sequence generated using the complex scrambling algorithm with input seed $f_{\text{PHY-HASH}}(2^{20} \times 51 + \text{SectorSeed})$ is denoted by s . The complex scrambling algorithm is described in 9.2.5.2 and SectorSeed is defined in 9.2.2.2.3. BPSK modulation is used to transmit the Forward Power Control Channel.

If PCVal = 0, then define $\beta = 1$. Otherwise, if PCVal = 1, then define $\beta = -1$. A value of $\sqrt{P_{\text{FPCCH}}} \times \beta \times s$ shall be transmitted on the subcarrier-symbol $(\text{FPCReportIndex} \bmod 3)$ of the Forward Link Control Segment resource index $R_{\text{FPCCH-BEGIN}} + \lfloor \text{FPCNodeIndex} \bmod 3 \rfloor$, provided that subcarrier-symbol is mapped to a Forward Link Control Segment available subcarrier.

9.4.1.2.4.9 Forward Shared Control Channel

The Forward Shared Control Channel (F-SCCH) carries control information for the Forward Data Channel transmission, as well as for group resource assignments.

9.4.1.2.4.9.1 Forward Shared Control Channel encoding

Each Forward Shared Control Channel packet shall be appended with a 16-bit CRC for all blocks except the GRA block, and with a 6-bit CRC for the GRA block. The FLCS MAC Protocol (see 8.5) indicates whether the block is a GRA block. The resulting sequence of bits denoted as $\{x_0, x_1, x_2, \dots\}$ shall be scrambled by a sequence $\{s_0, s_1, s_2, \dots\}$ generated using the common real scrambling algorithm with seed $f_{\text{PHY-HASH}}(2^{31} \times \alpha + 2048 \times \text{SectorSeed} + m \bmod 2048)$, where, m denotes either an access sequence index (in the case of Access Grant blocks) or a MACID (in the case of all other blocks) and is determined by the

FLCS MAC Protocol (8.5). In the seed for the hash functions, if m is an access sequence, then $\alpha = 0$ and if m is a MACID, then $\alpha = 1$. The scrambling operation shall comprise of flipping the bit x_i if $s_i = -1$.

The resulting sequence of bits shall be encoded using

- The rate-1/3 tail-biting convolutional code, if the SCCHTailBitingCodeEnabled parameter of the Active Set Management Protocol (11.7) is set to 1 and m is a unicast MACID.
- The rate-1/3 convolutional code, if the SCCHTailBitingCodeEnabled parameter of the Active Set Management Protocol (11.7) is set to 0 or m is not a unicast MACID.

The output of the encoder shall be channel interleaved, sequence-repeated, and converted to a sequence of $N_{\text{FSCCH, SUBCARRIERS}}$ modulation symbols using the procedure described in 9.2.6.1. No data scrambling shall be used for Forward Shared Control Channel packets.

The Forward Shared Control Channel packets that are transmitted in each Forward Link PHY frame are indexed by an ordered pair (a, b) where a takes values from 0 to $\text{MaxNumQPSKLABs} - 1$, and b takes values 0 and 1, where, MaxNumQPSKLABs is defined in 9.4.1.2.4.9.2. The Forward Shared Control Channel packets can be transmitted using either QPSK or 16-QAM modulation. b is set to 0 for all the Forward Shared Control Channel packets transmitted using QPSK modulation. For a given index a , the access network may either transmit a single packet using QPSK modulation with index $(a, 0)$, or two packets using 16-QAM modulation with indices $(a, 0)$ and $(a, 1)$. This implies that if a packet with index $(a, 0)$ is transmitted using QPSK modulation, then no packet with index $(a, 1)$ is transmitted. The FLCS MAC Protocol (see 8.5) determines the modulation format to be used for each Forward Shared Control Channel packet.

9.4.1.2.4.9.2 Forward Shared Control Channel modulation

Modulation for the Forward Shared Control Channel is described separately for the Common Segment and the LAB Segments, where the Common Segment and the LAB Segments are defined in 9.2.15.

The variable MaxNumQPSKLABs is defined by $N_{\text{SCCH-CS}} + N_{\text{FLCS-LAB-SEGMENTS}}N_{\text{SCCH-LAB}}$, where $N_{\text{SCCH-CS}}$ and $N_{\text{SCCH-LAB}}$ are defined in 9.4.1.2.4.9.2.1 and 9.4.1.2.4.9.2.2 respectively, and $N_{\text{FLCS-LAB-SEGMENTS}}$ is given by the NumLABSegments field of the Overhead Messages Protocol.

9.4.1.2.4.9.2.1 Modulation of Forward Shared Control Channel in the Common Segment

For each OFDM symbol, the notion of the Forward Shared Control Channel usable hop-ports is introduced and used in the remainder of this subclause. Define $R_{\text{SCCH-BEGIN}} = \lceil (N_{\text{FLCS-COMMON-BLOCKS}} \times 4 \times \text{MinSCCHResourceIndex} / (4 \times N_{\text{FLCS-COMMON-BLOCKS}})) \rceil$, where $\text{MinSCCHResourceIndex}$ is a field of the Overhead Messages Protocol. A hop-port shall be defined as unusable by the SCCH if one of the following conditions is satisfied:

- The hop-port is allocated to a Forward Link Control Segment resource with $R_{\text{FLCS}} < R_{\text{SCCH-BEGIN}}$.
- The hop-port is not mapped to a Forward Link Control Segment available subcarrier.

All other hop-ports are usable by the SCCH. Note that for $\text{UseDRCHForFLCS} = '0'$, if a hop-port location is unusable in one tile, it is unusable in all other tiles as well.

Define $N_{\text{SCCH-MODULATIONSYMBOLS}}$ to be the number of nominal modulation symbols in a QPSK-modulated Forward Shared Control Channel packet. $N_{\text{SCCH-MODULATIONSYMBOLS}}$ is given by the ModulationSymbolsPerQPSKLAB field of the Overhead Messages Protocol (see 11.6).

Define $N_{\text{SCCH-CS}}$ to be the number of Forward Shared Control Channel channel packet-pairs in the Common Segment. $N_{\text{SCCH-CS}}$ is determined according to the following procedure:

- $N_{\text{SCCH-CS}} = 0$ if $N_{\text{FLCS-LAB-SEGMENTS}} > 0$.
- $N_{\text{SCCH-CS}} = \left\lfloor \frac{(N_{\text{BLOCK}} N_{\text{FRAME}} N_{\text{FLCS-COMMON-BLOCKS}} - 3R_{\text{SCCH-BEGIN}})}{N_{\text{SCCH-MODULATIONSYMBOLS}}} \right\rfloor$ otherwise.

For $N_{\text{SCCH-CS}} > 0$, the Forward Shared Control Channel packets are then populated in the set of usable hop-ports in blocks $k = 0, 1, 2, \dots, N_{\text{FLCS-COMMON-BLOCKS}} - 1$ using the following rules:

- a) Initialize the hop-port counter i , block counter k , Orthogonal Frequency Division Multiplexing symbol counter j to 0.
- b) Initialize modulation symbol index $p(n) = 0$, for all $n = 0, 1, 2, \dots, N_{\text{SCCH-CS}} - 1$.
- c) If hop-port counter i is a Forward Shared Control Channel usable hop-port,
 - 1) Define $a = (k + j + i) \bmod N_{\text{SCCH-CS}}$.
 - 2) Define $b = 0$ if the Forward Shared Control Channel packet with index $(a, 0)$ is to be transmitted using QPSK modulation. Define $b = p(a) \bmod 2$ otherwise.
 - 3) Populate the modulation symbol with index $p(a)$ from Forward Shared Control Channel packet with index $(a, 0)$ on the i^{th} hop-port in the k^{th} hop-port block of the j^{th} OFDM symbol in the Common Segment if this packet is to be transmitted using QPSK modulation. Populate the modulation symbol with index $\lfloor p(a)/2 \rfloor$ from the Forward Shared Control Channel packet with index (a, b) on the i^{th} hop-port in the k^{th} hop-port block of the j^{th} OFDM symbol in the Common Segment if this packet is to be transmitted using 16 QAM modulation.
 - 4) The modulation symbol shall be modulated with power density P i.e., the value of the corresponding subcarrier shall be $\sqrt{P_s}$. The modulation shall be done on the tile-antenna with index 0. Note that the same power density P shall be used over all hop-ports assigned to this Forward Shared Control Channel packet. Different values of power density P may be used for different Forward Shared Control Channel packets. For the Forward Shared Control Channel packets transmitted using 16-QAM modulation, the power density P shall be equal to $P_{\text{Pilot}} R_{\text{T2P}}$, where, R_{T2P} is given by the 16QAMSCCHT2PRatio field of the Overhead Messages Protocol. When UseDRCHForFLCS is equal to '0', P_{Pilot} is the power density used to transmit the Forward Dedicated Pilot in the same tile. When UseDRCHForFLCS is equal to '1', P_{Pilot} is the power density used to transmit the Forward Common Pilot Channel. For the Forward Shared Control Channel packets transmitted using QPSK modulation. Determining the value of P is out of the scope of this standard.
 - 5) Increment $p(a)$ by 1.
- d) Increment i by 1. If $i = N_{\text{BLOCK}}$, set $k = k + 1$ and set $i = 0$.
- e) If $k \geq N_{\text{FLCS-COMMON-BLOCKS}}$, set $k = 0$ and increment $j = j + 1$.
- f) If $j \geq N_{\text{FRAME}}$, exit. Otherwise, go to step c).

9.4.1.2.4.9.2.2 Modulation of Forward Shared Control Channel in the LAB segments

For each OFDM symbol, the notion of the Forward Shared Control Channel usable hop-ports is introduced and used in the remainder of this subclause. A hop-port is defined as unusable by the Forward Shared Control Channel if it is not mapped to a Forward Link Control Segment available subcarrier.

Define $N_{\text{SCCH-LAB}}$ to be the number of Forward Shared Control Channel channel packets populated in each LAB Segment.

$N_{\text{SCCH-LAB}}$ is given by $N_{\text{SCCH-LAB}} = \left\lfloor \frac{3N_{\text{BLOCK}} N_{\text{FRAME}} N_{\text{FLCS}}}{N_{\text{SCCH-MODULATIONSYMBOLS}}} \right\rfloor$.

The Forward Shared Control Channel packets in the LAB segment q are then populated in the set of Forward Shared Control Channel usable hop-ports in hop-port blocks belonging to this LAB segment using the following rules:

- a) Initialize the hop-port counter i , block counter k , Orthogonal Frequency Division Multiplexing symbol counter j to 0.
- b) Initialize modulation symbol index $p(n) = 0$, for $n = q \times N_{\text{SCCH-LAB}}, q \times N_{\text{SCCH-LAB}} + 1, q \times N_{\text{SCCH-LAB}} + 2, \dots, q \times N_{\text{SCCH-LAB}} + N_{\text{SCCH-LAB}} - 1$.
- c) If hop-port counter i is a Forward Shared Control Channel usable hop-port,
 - 1) Define $a = (k + j + i) \bmod N_{\text{SCCH-LAB}} + q \times N_{\text{SCCH-LAB}}$.
 - 2) Define $b = 0$ if the Forward Shared Control Channel packet with index $(a, 0)$ is to be transmitted using QPSK modulation. Define $b = p(a) \bmod 2$ otherwise.
 - 3) Populate the modulation symbol with index $p(a)$ from the Forward Shared Control Channel packet with index $(a, 0)$ on the i^{th} hop-port in the k^{th} hop-port block of the j^{th} OFDM symbol in the LAB Segment q if this packet is to be transmitted using QPSK modulation. Populate the modulation symbol with index $\lfloor p(a)/2 \rfloor$ from the Forward Shared Control Channel packet with index (a, b) on the i^{th} hop-port in the k^{th} hop-port block of the j^{th} OFDM symbol in the LAB Segment q if this packet is to be transmitted using 16 QAM modulation.
 - 4) The modulation symbol shall be modulated with power density P i.e., the value of the corresponding subcarrier shall be $\sqrt{P_s}$. The modulation shall be done on the tile-antenna with index 0. Note that the same power density P shall be used over all hop-ports assigned to this Forward Shared Control Channel packet. Different values of power density P may be used for different Forward Shared Control Channel packets. For the Forward Shared Control Channel packets transmitted using 16-QAM modulation, the power density P shall be equal to $P_{\text{Pilot}} R_{\text{T2P}}$, where, R_{T2P} is given by the 16QAMSCCHT2PRatio field of the Overhead Messages Protocol. When UseDRCHForFLCS is equal to '0', P_{Pilot} is the power density used to transmit the Forward Dedicated Pilot in the same tile. When UseDRCHForFLCS is equal to '1', P_{Pilot} is the power density used to transmit the Forward Common Pilot Channel. For the Forward Shared Control Channel packets transmitted using QPSK modulation, the computation of the power density P is beyond the scope of this standard.
 - 5) Increment $p(a)$ by 1.
- d) Increment i by 1. If $i = N_{\text{BLOCK}}$, set $k = k+1$ and set $i = 0$.
- e) If $k \geq 3$, set $k = 0$ and increment $j = j + 1$.
- f) If $j \geq N_{\text{FRAME}}$, exit. Otherwise, go to step c).

9.4.1.2.5 Forward Data Channel

The Forward Data Channel (F-DCH) consists of one or more data packets that can span one or more Forward Link PHY frames. The set of Forward Link PHY frames on which the packets are transmitted is determined by the FTC MAC Protocol (8.5). Each data packet is also assigned a set of hop-ports in each PHY frame by the FTC MAC Protocol (8.5). For Forward Data Channel packets containing GRA bitmap, each data packet is associated with a packet size and modulation order assigned by the FTC MAC Protocol (8.5). For packets based on GRA bitmap, each packet is associated with packet a format index, spectral efficiency offset (for packet format indexes 2–4 only) and a modulation offset (for packet format indexes 2–3 only), which are also assigned by the FTC MAC Protocol (8.5). For all other Forward Data Channel packets, each packet is associated with packet format index assigned by the FTC MAC Protocol (8.5). In the following, power shall not be transmitted on an antenna with the DRCH structure or a tile antenna with the BRCH structure unless specifically specified.

9.4.1.2.5.1 Forward Data Channel Rotational Orthogonal Frequency Division Multiplexing

The Forward Data Channel shall use R-OFDM (see 9.2.9) if ROFDMEnabledFL is set to 1, where ROFDMEnabledFL is a parameter of the Active Set Management Protocol. If the Forward Data Channel uses rotational Orthogonal Frequency Division Multiplexing (see 9.2.9), it shall use it in the DRCH mode only. When the Forward Data Channel uses QPSK modulation, the dimension (D) of Rotational Orthogonal

Frequency Division Multiplexing shall be set to 4. When the Forward Data Channel uses 8-PSK, 16-QAM, or 64-QAM modulation, D shall be set to 2.

The optimal rotational angles corresponding to the various packet formats are summarized in Table 220. Note that for the third to sixth transmissions, this angle shall be set to zero.

Table 220—Optimal Rotational Angle for Rotational Orthogonal Frequency Division Multiplexing

Packet Format index	Optimal angle	
	First transmission	Second transmission
0	0	0
1	0	0
2	$0.3 \times \pi/4$	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	$0.2 \times \pi/4$	0
10	$0.3 \times \pi/4$	0
11	$0.7 \times \pi/4$	0
12	0	0
13	0	0
14	$0.3 \times \pi/4$	$0.3 \times \pi/4$
15	0	0

If the number of modulation symbols in a packet is denoted by N_{ModSym} , then the last ($N_{\text{ModSym}} \bmod D$) symbols shall not be rotated. The remaining symbols shall be rotated in groups of D as specified in 9.2.9, and assigned to hop-ports as specified in the remainder of 9.4.1.2.5.

9.4.1.2.5.2 Forward Data Channel Packet Data Control assignment block assignments

When the access terminal assignment (ATA) as specified by the FTC MAC Protocol comprises of hop-port blocks allocated to the Forward Link Control Segment and Packet Data Control Assignment Block is enabled at the access network ($\text{PDCABResourceSharingEnabled} = 1$, see 11.6), the access network shall perform two operations:

- Transmit the data packet as described in 9.4.1.2.5.4 through 9.4.1.2.5.8 on those hop-ports of the ATA that are not allocated to the Forward Link Control Segment.
- Transmit the same data packet using the inverted sequence repetition operation on the hop-ports of the ATA that are allocated to the Forward Link Control Segment.

9.4.1.2.5.3 Forward Data Channel available subcarriers

Not all subcarriers may be available for modulation by the Forward Data Channel. For example, subcarriers in which the Forward Dedicated Pilot Channel is transmitted cannot be used by the Forward Data Channel. In this subclause, the notion of Forward Data Channel unavailable subcarriers is defined. A subcarrier is unavailable for the Forward Data Channel if:

- The subcarrier is a pilot subcarrier i.e., it is allocated to one of the Forward Link Pilot Channels (Forward Dedicated Pilot Channel, Forward Channel Quality Indicator Pilot Channel, Forward Common Pilot Channel).
- The subcarrier is part of a BeaconOnlyOFDMSymbol.
- The subcarrier is allocated of the Forward Link Control Segment and is not explicitly defined as being available as part of the Packet Data Control Assignment Block.

Furthermore, if ResourceChannelMuxMode = 1, all subcarriers allocated to the DRCH are not available when modulating BRCH resources.

All subcarriers that are not unavailable as defined above shall be referred to as “Forward Data Channel available subcarriers.”

9.4.1.2.5.4 Forward Data Channel Single Input Single Output Mode

On the Forward Data Channel, the Single Input Single Output mode deals with single effective antenna at the access network. The access terminal may use receive diversity if it chooses to use multiple antennas.

9.4.1.2.5.4.1 Forward Data Channel Single Input Single Output Mode Data Packet Encoding

The Forward Data Channel packet is generated by the FTC MAC Protocol (see 8.5), and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2.6.1. A CRC length of $N_{\text{CRC,Data}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(\text{SectorSeed} + m \times 2^{20})$ shall be used for the data scrambling operation, where SectorSeed is defined in 9.2.2.2.3, and m denotes the MACID of the access terminal of interest except in the case of multicast group resource transmissions. For the case of multicast group resource transmissions, m denotes the GroupID.

9.4.1.2.5.4.2 Forward Data Channel Single Input Single Output Mode Data Packet Transmission

The data packet shall be modulated on to the hop-ports assigned to this packet according to the following procedure:

- a) Initialize a port counter i , a HARQ retransmission counter r , a frame counter f , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- b) Let $F(r)$ be the total number of PHY frames to be used in the r^{th} HARQ retransmission of the packet, as specified by the FTC MAC Protocol (8.5). The frames shall be indexed $(r, 0), (r, 1) \dots (r, F(r) - 1)$.
- c) Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) . Let the resulting sequence be denoted by $p_0, p_1, \dots, p_{n-1}$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .
- d) Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} Orthogonal Frequency Division Multiplexing symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY frame (r, f) , which is a function of the packet format and HARQ retransmission index r . If n_{sc} is a Forward Data Channel available subcarrier, then a modulation symbol s from subpacket m with modulation order q is generated by the modulator according to the procedure described in 9.2.7.3. Here m shall be equal to $(i_{\text{TILE}} + (j + i \bmod N_{\text{BLOCK}}) \bmod N_{\text{SUBPACKETS-IN-TILE}}) \bmod t$, where t is

the total number of subpackets in the packet (equal to $N_{\text{DCH, SUBPACKETS}}$), N_{BLOCK} is the number of subcarriers in a block, $i_{\text{TILE}} = \lfloor i/N_{\text{BLOCK}} \rfloor$, and $N_{\text{SUBPACKETS-IN-TILE}}$ is computed as follows:

$$N_{\text{SUBPACKETS-IN-TILE}} = t \text{ if } i_{\text{TILE}} < (N_{\text{TILES}} \bmod t), \text{ where } N_{\text{TILES}} = \lfloor n/N_{\text{BLOCK}} \rfloor$$

$$N_{\text{SUBPACKETS-IN-TILE}} = \min \left(t, \left\lceil \frac{8t}{N_{\text{TILES}} - (N_{\text{TILES}} \bmod t)} \right\rceil \right) \text{ otherwise.}$$

- e) The modulation symbol s shall be modulated with power density P on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{P}s$. The modulation shall be done on the antenna with index 0 if i_{TILE} is a DRCH resource, and on the tile-antenna with index 0 if i_{TILE} is a BRCH resource. The same power density P shall be used over all DRCH hop-ports assigned to this packet. Different values of power density P may be used for different BRCH resources. Determining the value of P is out of the scope of this standard.
- f) Increment i . If $i = n$, increment j and set $i = 0$.
- g) If $j = N_{\text{FRAME}}$, set $j = 0$ and increment f .
- h) If $f = F(r)$, then increment r and set $f = 0$.
- i) If the last HARQ transmission has been completed (as determined by the FTC MAC Protocol), then stop. Else, repeat steps c) through h).

9.4.1.2.5.5 Forward Data Channel Precoding for Multiple Input Multiple Output Mode

If precoding is used on the Forward Data Channel, the tile antennas used for Multiple Input Multiple Output or Space Time Transmit Diversity transmissions are obtained from the effective antennas through the use of precoding matrices as described in 9.2.8.2. When precoding is used by the access network, these tile antennas shall be used for the Space Time Transmit Diversity, Multi-Codeword and Single Codeword modes as described in 9.4.1.2.5.6, 9.4.1.2.5.7, and 9.4.1.2.5.8.

With the Knockdown precoder in ResourceChannelMuxMode 1, Knockdown precoder in DRCH in ResourceChannelMuxMode 2 and any precoder in BRCH in ResourceChannelMuxMode 2, the access network can choose to use any precoding matrix and vectors. With the Readymade precoder in ResourceChannelMuxMode 1 or Readymade precoder in DRCH in ResourceChannelMuxMode 2, the access network shall use the precoding matrix that was reported by the access terminal in its latest message sent on the reverse beam feedback channel that is transmitted in the Reverse Orthogonal Frequency Division Multiple Access Dedicated Control Channel by the access terminal.

9.4.1.2.5.6 Forward Data Channel Space Time Transmit Diversity Mode

In the Space Time Transmit Diversity (STTD) mode, the number of effective antennas used can be two or four. In the four antenna case, there are two modes of Space Time Transmit Diversity that are supported by the access terminal. When the number of effective antennas used is four, the Space Time Transmit Diversity mode may be used in conjunction with Antenna Selection.

9.4.1.2.5.6.1 Forward Data Channel Data Packet Encoding for Space Time Transmit Diversity Mode

In the Space Time Transmit Diversity mode, each Forward Data Channel packet is generated by the FTC MAC Protocol, and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2.6.1. A CRC length of $N_{\text{CRC,Data}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(\text{SectorSeed} + m \times 2^{20})$ shall be used for the data scrambling operation, where SectorSeed is defined in 9.2.2.2.3 and m denotes the MACID of the access terminal of interest except

in the case of multicast group resource transmissions. For the case of multicast group resource transmissions, m denotes the GroupID.

9.4.1.2.5.6.2 Forward Data Channel Space Time Transmit Diversity Modes

Space Time Transmit Diversity can be supported for two or four antennas for a sequence of NSTTD modulation symbols. Let M denote the number of antennas.

For the two antenna case, each set of two modulation symbols (S_0, S_1) shall be mapped to a 2×2 matrix $Y = A$, shown in Figure 127. $Y_{k,i}$ denotes the i^{th} modulation symbol to be transmitted on the k^{th} antenna. Modulation vectors Y_0 and Y_1 refer to the columns of the matrix Y . In this case, define $N_{\text{STTD}} = 2$.

For the four antenna case, two modes are defined. The following matrices are defined:

$$A = \begin{bmatrix} S_i & -S_{i+1}^* & 0 & 0 \\ S_{i+1} & S_i^* & 0 & 0 \\ 0 & 0 & S_{i+2} & -S_{i+3}^* \\ 0 & 0 & S_{i+3}^* & S_{i+2} \end{bmatrix}, \quad B = \begin{bmatrix} S_i & -S_{i+1}^* & S_{i+4} & -S_{i+6}^* \\ S_{i+1} & S_i^* & S_{i+5} & -S_{i+7}^* \\ S_{i+2} & -S_{i+3}^* & S_{i+6} & S_{i+4}^* \\ S_{i+3} & S_{i+2}^* & S_{i+7} & S_{i+5}^* \end{bmatrix}.$$

In Space Time Transmit Diversity Mode A, three circulation matrices are defined as follows:

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

In Space Time Transmit Diversity Mode A, for each set of four modulation symbols (S_0, S_1, S_2, S_3), the matrix $Y = CA$ shall denote the output of the four antennas as illustrated in Figure 128, where $C \in \{C_1, C_2, C_3\}$. Matrices in C shall be used in a circular fashion for sets of four modulation symbols, starting with the first matrix in each PHY frame. In this case, define $N_{\text{STTD}} = 4$. $Y_{k,i}$ denotes the i^{th} modulation symbol to be transmitted on the k^{th} antenna. Modulation vectors Y_0, Y_1, Y_2 , and Y_3 refer to the columns of the matrix Y .

In Space Time Transmit Diversity Mode B, six circulation matrices are defined as follows:

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$C_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad C_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad C_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

for each set of eight modulation symbols ($S_0, S_1, \dots, S_7$), the matrix $Y = CB$ shall denote the output of the four antennas as illustrated in Figure 128, where $C \in \{C_1, C_2, C_3, C_4, C_5, C_6\}$. Matrices in C shall be

used in a circular fashion for sets of eight modulation symbols, starting with the first matrix for each PHY frame. In this case, define $N_{\text{STTD}} = 8$. $Y_{k,i}$ denotes the i^{th} modulation symbol to be transmitted on the k^{th} antenna. Modulation vectors $\mathbf{Y}_0$, $\mathbf{Y}_1$, $\mathbf{Y}_2$, and $\mathbf{Y}_3$ refer to the columns of the matrix $\mathbf{Y}$.

9.4.1.2.5.6.3 Forward Data Channel Data Packet Transmission for Space Time Transmit Diversity Mode

The data packets shall be modulated on to the hop-ports assigned to this packet according to the following procedure:

- Initialize a port counter i , a HARQ transmission counter r , a frame counter f , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- Create an empty transmission queue Q .
- Let $F(r)$ be the total number of PHY frames to be used by the r^{th} HARQ retransmission of the packet, as specified by the FTC MAC Protocol (see 8.5). The frames shall be indexed $(r, 0)$, $(r, 1) \dots (r, F(r) - 1)$.
- Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) in increasing order. Let the resulting sequence be denoted by the n hop-ports $(p_0, p_1, \dots, p_{n-1})$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .
- If the transmission queue Q is empty, then a sequence of N_{STTD} modulation symbols $\{s_0, s_1, \dots, s_{N_{\text{STTD}}-1}\}$ from subpacket m respectively with modulation order q is generated by the modulator according to the procedure described in 9.2.6, where the subpacket m shall be equal to $(i_{\text{TILE}} + (j + i \bmod N_{\text{BLOCK}}) \bmod N_{\text{SUBPACKETS-IN-TILE}}) \bmod t$, where t is the total number of subpackets in the packet (equal to $N_{\text{DCH, SUBPACKETS}}$), N_{BLOCK} is the number of subcarriers in a block, $i_{\text{TILE}} = \lfloor i / N_{\text{BLOCK}} \rfloor$ and $N_{\text{SUBPACKETS-IN-TILE}}$ is computed as follows:

$$N_{\text{SUBPACKETS-IN-TILE}} = t \text{ if } i_{\text{TILE}} < (N_{\text{TILES}} \bmod t), \text{ where } N_{\text{TILES}} = \lfloor n / N_{\text{BLOCK}} \rfloor$$

$$N_{\text{SUBPACKETS-IN-TILE}} = \min \left(t, \left\lceil \frac{16t}{N_{\text{TILES}} - (N_{\text{TILES}} \bmod t)} \right\rceil \right) \text{ otherwise.}$$

The sequence of modulation symbols $\{s_0, s_1, \dots, s_{N_{\text{STTD}}-1}\}$ is used to generate a sequence of modulation vectors $\{\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_{M-1}\}$ as specified in 9.4.1.2.5.6.2. The modulation vectors $\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_{M-1}$ are inserted into the transmission queue Q , in that order.

- Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} OFDM symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY frame (r, f) , which is a function of the packet format and HARQ transmission index r . If n_{sc} is a F-DCH available subcarrier in the j^{th} or $(j+1)^{\text{th}}$ OFDM symbol, then let $\mathbf{V}_0$ and $\mathbf{V}_1$ denote the modulation vectors at the head of the transmission queue Q , and let the modulation symbols $V_{k,0}$ and $V_{k,1}$ denote their k^{th} component. Remove the modulation vectors $\mathbf{V}_0$ and $\mathbf{V}_1$ from the transmission queue Q .
- If the subcarrier n_{sc} is a F-DCH available subcarrier in both the j^{th} and $(j+1)^{\text{th}}$ OFDM symbols, then all the modulation symbols on subcarrier n_{sc} and symbols j and $(j+1)$ shall be assigned the same energy spectral density P/M . If the subcarrier n_{sc} is not a F-DCH available subcarrier in the j^{th} OFDM symbol, every odd occurrence of a non-zero symbol of $\mathbf{V}_1$, starting with the first non-zero symbol of the modulation vector $\mathbf{V}_1$ is assigned energy spectral density of $2P/M$, and all other modulation symbols of vectors $\mathbf{V}_0$ and $\mathbf{V}_1$ are assigned energy spectral density of zero. If the subcarrier n_{sc} is not a F-DCH available subcarrier in the $(j+1)^{\text{th}}$ OFDM symbol, then every odd occurrence of a non-zero symbol of $\mathbf{V}_0$, starting with the first non-zero symbol of the modulation vector $\mathbf{V}_0$ is assigned energy spectral density of $2P/M$, and all other modulation symbols of vectors $\mathbf{V}_0$ and $\mathbf{V}_1$ are assigned energy spectral density of zero.

- h) For $k = 0, 1, \dots, M - 1$, the modulation symbol $V_{k,0}$ shall be transmitted with its assigned energy spectral density on hop-port p_i , antenna index k and OFDM symbol j of the PHY frame (r,f) . For $k = 0, 1, \dots, M - 1$, the modulation symbol $V_{k,1}$ shall be transmitted with its assigned energy spectral density on hop-port p_i , antenna index k and OFDM symbol $(j + 1)$ of the PHY frame (r,f) .
- i) Increment i . If $i = n$, set $i = 0$ and increment j by 2.
- j) If $j = N_{\text{FRAME}}$, set $j = 0$ and increment f .
- k) If $f = F(r)$, then increment r and set $f = 0$.
- l) If the last HARQ transmission has been completed [as determined by the FTC MAC Protocol (see 8.5)], then stop. Else repeat steps b) through k).

9.4.1.2.5.7 Forward Data Channel Multiple Input Multiple Output Multi-Codeword Mode

Multiple data packets may be transmitted in Multiple Input Multiple Output Multi-Codeword mode. The number of packets is equal to N_{LAYERS} , the number of layers for this transmission as specified by the FTC MAC Protocol (8.5). The layers shall be indexed 0 through $N_{\text{LAYERS}} - 1$. A separate packet shall be transmitted on each layer.

9.4.1.2.5.7.1 Forward Data Channel Permutation Matrices for Multi-Codeword Multiple Input Multiple Output Mode

Let $P_p^{\text{NUM\_LAYER}}$ denote the set of all permutation matrices of order NUM_LAYER ($p = 0, 1, \dots, \text{NUM\_LAYER}! - 1$). The set of all such matrices for NUM_LAYER = 1, 2, 3, and 4 are enumerated in 9.2.8.3. All the permutation matrices shall be used.

9.4.1.2.5.7.2 Forward Data Channel Data Packet Encoding for Multiple Input Multiple Output Mode

Each Forward Data Channel packet is generated by the FTC MAC Protocol, and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2.6. A CRC length of $N_{\text{CRC,Data}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(\text{SectorSeed} + m \times 2^{20} + k \times 2^{31} + 2^{33})$ shall be used for the data scrambling operation, where SectorSeed is as defined in 9.2.2.2.3 and m denotes either the MACID of the access terminal of interest (for unicast transmission) or the broadcast MACID (for broadcast transmission).

9.4.1.2.5.7.3 Forward Data Channel Data Packet Transmission for Multi-Codeword Multiple Input Multiple Output Mode

The N_{LAYERS} data packets shall be modulated on to the hop-ports assigned to this packet according to the following procedure:

- a) Initialize a port counter i , a HARQ retransmission counter r , a frame counter f , a permutation counter p , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- b) Let $F(r)$ be the total number of PHY frames to be used by the r^{th} HARQ transmission of the packet, as specified by the FTC MAC Protocol. The frames shall be indexed $(r, 0), (r, 1) \dots (r, F(r) - 1)$.
- c) Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) in increasing order. Let the resulting sequence be denoted by $p_0, p_1, \dots, p_{n-1}$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .
- d) Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} Orthogonal Frequency Division Multiplexing symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY frame (r, f) , which is a function of the packet format and HARQ transmission index r . If n_{sc} is not a pilot subcarrier and is a Forward Data Channel available subcarrier, then a sequence of N_{LAYERS} modulation symbols $\{s_0, s_1, \dots, s_{N_{\text{LAYERS}}-1}\}$ from subpackets $\{m_0, m_1, \dots, m_{N_{\text{LAYERS}}-1}\}$ respectively with modulation order q is generated by the modulator according to the procedure

described in 9.2.6, where the subpacket m_k of the data packet on layer k shall be equal to $(i_{\text{TILE}} + (j + i \bmod N_{\text{BLOCK}}) \bmod N_{\text{SUBPACKETS-IN-TILE}}) \bmod t$, where t is the total number of subpackets in the packet (equal to $N_{\text{DCH, SUBPACKETS}}$ for that layer), N_{BLOCK} is the number of subcarriers in a block, $i_{\text{TILE}} = \lfloor i / N_{\text{BLOCK}} \rfloor$ and $N_{\text{SUBPACKETS-IN-TILE}}$ is computed as follows:

$$N_{\text{SUBPACKETS-IN-TILE}} = t \text{ if } i_{\text{TILE}} < (N_{\text{TILES}} \bmod t), \text{ where } N_{\text{TILES}} = \lfloor n / N_{\text{BLOCK}} \rfloor$$

$$N_{\text{SUBPACKETS-IN-TILE}} = \min \left(t, \left\lceil \frac{16t}{N_{\text{TILES}} - (N_{\text{TILES}} \bmod t)} \right\rceil \right) \text{ otherwise.}$$

- e) This modulation symbol s_k ($k = 0, 1, \dots, N_{\text{LAYERS}} - 1$) shall be modulated with power density P on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{P}s_k$. The same power density P shall be used over all DRCH hop-ports assigned to this packet. Different values of power density P may be used for different BRCH resources.
- f) Define $y_k = \sqrt{P}s_k$. Let $\mathbf{Y}$ be the vector $\{y_k\}$, $k = 0, 1, \dots, N_{\text{LAYERS}} - 1$. Define $z_k = (P_{\text{p}}^{\text{NUM\_LAYER}} \mathbf{Y})_k$.
- g) z_k shall be transmitted on the antenna with index k .
- h) Increment i . If $i = n$, increment j and set $i = 0$.
- i) Increment p . If $p = (N_{\text{LAYERS}})!$, set $p = 0$.
- j) If $j = N_{\text{FRAME}}$, set $j = 0$ and increment f .
- k) If $f = F(r)$, then increment r and set $f = 0$.
- l) If the last HARQ transmission has been completed [as determined by the FTC MAC Protocol (8.5)], then stop. Else repeat steps b) through h).

9.4.1.2.5.8 Forward Data Channel Multiple Input Multiple Output Single Codeword mode

A single data packet shall be transmitted in Multiple Input Multiple Output Single Codeword mode.

9.4.1.2.5.8.1 Forward Data Channel Data Packet Encoding for Multiple Input Multiple Output Single Codeword mode

The Forward Data Channel packet is generated by the FTC MAC Protocol (8.5), and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2.6. A CRC length of $N_{\text{CRC,Data}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(\text{SectorSeed} + m \times 2^{20})$ shall be used for the data scrambling operation, where SectorSeed is as defined in 9.2.2.2.3 and m denotes the MACID of the access terminal of interest.

9.4.1.2.5.8.2 Forward Data Channel Data Packet Transmission for Multiple Input Multiple Output Single Codeword mode

The data packet shall be modulated on to the hop-ports assigned to this packet according to the following procedure:

- a) Initialize a port counter i , a HARQ retransmission counter r , a frame counter f , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- b) Let $F(r)$ be the total number of PHY frames to be used by the r^{th} HARQ transmission of the packet, as specified by the FTC MAC Protocol. The frames shall be indexed $(r, 0), (r, 1) \dots (r, F(r) - 1)$.
- c) Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) . Let the resulting sequence be denoted by $p_0, p_1, \dots, p_{n-1}$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .
- d) Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} Orthogonal Frequency Division Multiplexing symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY

frame (r, f) , which is a function of the packet format and HARQ retransmission index r . If n_{sc} is a Forward Data Channel available subcarrier, then a sequence of N_{LAYERS} modulation symbols $\{s_0, s_1, \dots, s_{N_{LAYERS}-1}\}$ from subpacket m with modulation order q is generated by the modulator according to the procedure described in 9.2.6. Here m shall be equal to $(i_{TILE} + (j + i \bmod N_{BLOCK}) \bmod N_{SUBPACKETS-IN-TILE}) \bmod t$. Here t is the total number of subpackets in the packet (equal to $N_{DCH, SUBPACKETS}$), N_{BLOCK} is the number of subcarriers in a block, $i_{TILE} = \lfloor i / N_{BLOCK} \rfloor$, and $N_{SUBPACKETS-IN-TILE}$ is computed as follows:

$$N_{SUBPACKETS-IN-TILE} = t \text{ if } i_{TILE} < (N_{TILES} \bmod t), \text{ where } N_{TILES} = \lfloor n / N_{BLOCK} \rfloor$$

$$N_{SUBPACKETS-IN-TILE} = \min \left(t, \left\lceil \frac{16t}{N_{TILES} - (N_{TILES} \bmod t)} \right\rceil \right) \text{ otherwise.}$$

- e) This modulation symbol s_k ($k = 0, 1, \dots, N_{LAYERS} - 1$) shall be modulated with power density P on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{P}s_k$. The modulation shall be done on the antenna with index k if i_{TILE} is a DRCH resource, and on the tile-antenna with index k if i_{TILE} is a BRCH resource. The same power density P shall be used over all DRCH hop-ports assigned to this packet. Different values of power density P may be used for different BRCH resources. Determining the value of P is out of the scope of this standard.
- f) Increment i . If $i = n$, increment j and set $i = 0$.
- g) If $j = N_{FRAME}$, set $j = 0$ and increment f .
- h) If $f = F(r)$, then increment r and set $f = 0$.
- i) If the last HARQ transmission has been completed [as determined by the FTC MAC Protocol (8.5)], then stop. Else repeat steps b) through h).

9.4.1.3 Limitations on emissions

9.4.1.3.1 Conducted spurious emissions

The system shall meet all applicable regulatory and legal requirements.

9.4.1.3.2 Radiated spurious emissions

The system shall meet all applicable regulatory and legal requirements.

9.4.1.4 Synchronization, timing, and phase

9.4.1.4.1 Timing reference source

Each sector shall use a time-base reference from which all time-critical transmission components, including superframe boundaries, PHY frame boundaries, and superframe indices, shall be derived. For synchronous systems, this shall be related to the System Time as outlined in 9.2.2. In asynchronous mode, there is no requirement for the alignment of the time-base references of two sectors.

9.4.1.4.2 Sector transmission time

Each sector shall radiate the superframe boundary aligned to its time-base reference. Time measurements are made at the sector antenna connector. If a sector has multiple radiating antenna connectors for the same channel, time measurements are made at the antenna connector having the earliest radiated signal.

The rate of change for timing corrections shall not exceed 102 nanoseconds (ns) per 200 milliseconds (ms).

9.4.1.5 Transmitter performance requirements

9.4.1.5.1 Frequency control requirements

At the Access Node, the transmitted center frequency, and the symbol clock frequency should be derived from the system time (see 5.4). At the AN, the reference frequency accuracy shall be better than $\pm 1\%$ of the subcarrier spacing.

9.4.1.6 Transmit timing synchronization

At the AN, in Wideband TDD mode, the AN shall synchronize its burst timing within 10 microseconds of the system timing (GPS).

Burst timing synchronization does not apply to FDD mode.

9.4.1.7 AN transmit waveform quality

The goal is to make sure that the receiver at AT does not degrade by more than 0.5 dB in performance due to transmitter imperfections.

9.4.1.7.1 Procedure for AN MSCE measurement

The following procedure shall be followed:

- a) The AN is assigned all tiles for data except for those designated for guard band.
- b) The DC tone shall not be populated.
- c) The signal is transmitted from the first effective antenna and is cable-connected to the receiver with two receive antennas. Denote the received samples on antennas 1 and 2 by y_1 and y_2 respectively.
- d) After down conversion, the receiver performs time and frequency offsets calculation.
- e) Time and frequency correction shall be applied to the received signal yielding to samples z_1 and z_2 from antenna 1 and antenna 2 respectively.
- f) Using pilots, for each data tone, estimate the average channel per Rx antenna, denote the estimates by h_1 and h_2 for antenna 1 and antenna 2 respectively.
- g) The actual received constellation point on that tone is given by $\hat{x} = \frac{(h_1^* z_1 + h_2^* z_2)}{(|h_1|^2 + |h_2|^2)}$

where a^* is the complex conjugate of a and $|a|$ is the absolute value of a .

- h) Find the closest constellation point to $\hat{x}$ in Euclidean distance and denote that point by x .
- i) Find the MSCE metric given by

$$\epsilon = \frac{\sum_{j=1}^{N_f} \sum_{k=1}^{N_p} (x_I(j, k) - \hat{x}_I(j, k))^2 + (x_Q(j, k) - \hat{x}_Q(j, k))^2}{\sum_{j=1}^{N_f} \sum_{k=1}^{N_p} (x_I(j, k))^2 + (x_Q(j, k))^2}$$

where a_I, a_Q are the real and imaginary parts of a , N_p is the number of modulation symbols in the packet and N_f is the number of frames used for averaging MSCE.

The MSCE shall not exceed the values in Table 221.

Table 221—AN MSCE values as a function of modulation order

(Packet Format) PF	Modulation	MSCE (dB)
2 – 1st transmission	QPSK	–15
4 – 2nd transmission	8-PSK	–18
4 – 1st transmission	16-QAM	–21
6 – 1st transmission	64-QAM	–26

9.4.2 Receiver

9.4.2.1 Channel spacing and designation

Channel spacing and designations for the access network reception shall be as specified in 9.3.1.1.1.

9.4.2.2 Demodulation characteristics

The access network demodulation process shall perform complementary operations to the access terminal modulation process on the Reverse Link.

9.4.2.3 Limitations on emissions

The access network shall meet the requirements in 3.6 of 3GPP2 C.S0010-C.

9.4.2.4 Receiver performance requirements

9.4.2.4.1 Receiver sensitivity requirements

The AN receiver sensitivity is as defined in 9.3.2. The low and the high sensitivity levels shall not exceed the values in Table 214 and Table 215, respectively.

9.5 Broadcast Multicast Services (BCMCS)

This subclause defines requirements specific to access network equipment and operation for the support of Broadcast and Multicast Services (BCMCS). It also describes the optional supercast operation of unicast traffic on the broadcast portion.

9.5.1 Broadcast and Multicast Services transmitter

The transmitter shall reside in each sector of the access network. These requirements apply to the transmitter in each sector.

9.5.1.1 Frequency parameters

9.5.1.1.1 Channel spacing and designation

See 3GPP2 C.S0057-B for a description of the band classes that an access network may support.

9.5.1.1.2 Frequency tolerance

The access network shall meet the requirements in 9.4.1.6.

9.5.1.2 Power output characteristics

The access network shall meet the requirements in 3GPP2 C.S0010-C.

9.5.1.3 Modulation characteristics

Two radio configurations are defined for the BCMCS services, which are described in 9.5.1.3.1.2.1 and 9.5.1.3.1.2.2, respectively.

9.5.1.3.1 BCMCS signals

The BCMCS channels are described in Table 222.

Table 222—Description of the BCMCS Channels

F-BMPICH	Forward Broadcast and Multicast Pilot Channel
F-BCMCSCH	Forward Broadcast and Multicast Services Channel

9.5.1.3.1.1 Channel Structures

The structure of the Forward Broadcast and Multicast Services Channel is shown in Figure 137. The channel structure for the single effective antenna case is shown in Figure 138 and Figure 139.

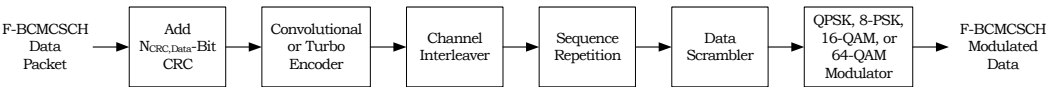


Figure 137—Forward Broadcast and Multicast Services Channel Structure

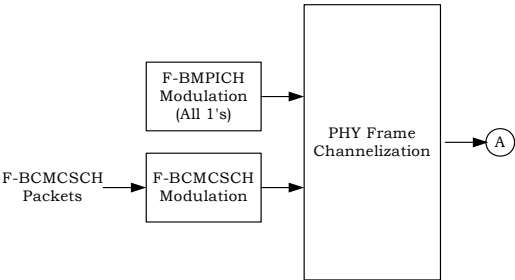


Figure 138—Channel Structure in the PHY frames

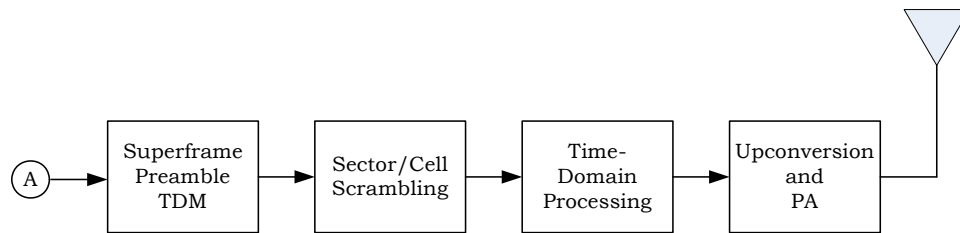


Figure 139—Channel Structure for the Single-Transmit-Antenna Case

9.5.1.3.1.2 Modulation Parameters for the Forward Broadcast and Multicast Services Channel

The Forward Broadcast and Multicast Services Channel uses the Orthogonal Frequency Division Multiplexing symbol parameters specified in Table 223 for Radio Configuration 1 and the Orthogonal Frequency Division Multiplexing symbol parameters specified in Table 224. for Radio Configuration 2. If the Forward Broadcast and Multicast Services Channel uses three transmissions per packet, the last subpacket shall use the Orthogonal Frequency Division Multiplexing symbol parameters specified in Table 207.

9.5.1.3.1.2.1 Radio Configuration 1

The Forward Broadcast and Multicast Services Channel shall be transmitted using Orthogonal Frequency Division Multiplexing. The Orthogonal Frequency Division Multiplexing symbols shall be transmitted in a superframe structure where each superframe consists of a superframe preamble followed by a number of PHY frames. The superframe preamble and PHY frames contain contiguous groups of Orthogonal Frequency Division Multiplexing symbols. An Orthogonal Frequency Division Multiplexing symbol consists of $N_{\text{FFT},B}$ individually modulated subcarriers that carry complex-valued data. Complex-valued data are represented in the form $d = (d_{\text{re}}, d_{\text{im}})$, where d_{re} and d_{im} represent the real and imaginary components, respectively. The subcarriers in each Orthogonal Frequency Division Multiplexing symbol are numbered from 0 through $N_{\text{FFT},B} - 1$.

The Orthogonal Frequency Division Multiplexing symbol parameters shall be as specified in Table 223.

Table 223—BCMS OFDM symbol numerology for Radio Configuration 1

Parameter	Value	Units
Chip Rate ($1/T_{\text{CHIP}}$)	1.2288	Mcps
$N_{\text{FFT},B}$	128	
Subcarrier Spacing	9.6	kHz
Cyclic Prefix Duration of OFDM Symbols in the BCMCS PHY frames($T_{\text{CP},B}$)	22.78, 30.22, 37.66, or 45.1	μs
Windowing Guard Interval ($T_{\text{WGI},B}$)	3.26	μs
Number of OFDM Symbols in the BCMCS PHY frames	7	

9.5.1.3.1.2.2 Radio Configuration 2

The Forward Broadcast and Multicast Services Channel uses Orthogonal Frequency Division Multiplexing. The Orthogonal Frequency Division Multiplexing symbols shall be transmitted in a superframe structure where each superframe consists of a superframe preamble followed by a number of PHY frames. The superframe preamble and PHY frames contain contiguous groups of Orthogonal Frequency Division Multiplexing symbols. An Orthogonal Frequency Division Multiplexing symbol consists of $N_{\text{FFT},B}$ individually modulated subcarriers that carry complex-valued data. Complex-valued data are represented in the form $d = (d_{\text{re}}, d_{\text{im}})$, where d_{re} and d_{im} represent the real and imaginary components, respectively. The subcarriers in each Orthogonal Frequency Division Multiplexing symbol are numbered from 0 through $N_{\text{FFT},B} - 1$.

The Orthogonal Frequency Division Multiplexing symbol parameters shall be as specified in Table 224.

Table 224—Orthogonal Frequency Division Multiplexing symbol numerology for Radio Configuration 2

Parameter	Value	Units
Chip Rate ($1/T_{\text{CHIP}}$)	1.2288	Mcps
$N_{\text{FFT},B}$	320	
Subcarrier Spacing	3.8	kHz
Cyclic Prefix Duration of OFDM Symbols in the BCMCS PHY frames($T_{\text{CP},B}$)	39.67, 57.03, 74.39, or 91.75	μs
Windowing Guard Interval ($T_{\text{WGI},B}$)	3.26	μs
Number of OFDM Symbols in the BCMCS PHY frames	3	

9.5.1.3.1.3 Forward Broadcast and Multicast Pilot Channel

The Forward Broadcast and Multicast Pilot Channel (F-BMPICH) is composed of uniformly distributed pilot subcarriers within each Orthogonal Frequency Division Multiplexing symbol, with staggering between continuous Orthogonal Frequency Division Multiplexing symbols. In this subclause, the pilot location is defined as if all the subcarriers of all the Orthogonal Frequency Division Multiplexing symbols within a superframe are used for broadcasting purpose. However, pilot insertion is only done within the subcarriers that are used for BCMCS.

For pilot insertion purpose, all Orthogonal Frequency Division Multiplexing symbols in a superframe are labeled sequentially. Let j be Orthogonal Frequency Division Multiplexing symbol index within a superframe and i be subcarrier index within each Orthogonal Frequency Division Multiplexing symbol. Subcarrier index $i = 0$ to $N_{\text{FFT},B} - 1$. Orthogonal Frequency Division Multiplexing symbol index $j = 0$ to 167 for Radio Configuration 1 and $j = 0$ to 71 for Radio Configuration 2.

The i^{th} tone in Orthogonal Frequency Division Multiplexing symbol j is a pilot locations if and only if

$$i \bmod 8 = (S[j] + \text{PilotStagger}) \bmod 8$$

where $S[j] = 0$ if j is even and $S[j] = 4$ if j is odd. The definition of PilotStagger is given in 14.2.2.3.4.1.1.

A power boost is applied to pilot subcarriers to improve the performance of channel estimation. The default traffic subcarriers to pilot subcarrier power ratio is specified in 14.2, and can be modified by Overhead Messages.

9.5.1.3.2 Forward Broadcast and Multicast Services Channel

The Forward Broadcast and Multicast Services Channel (F-BCMCSCH) is an encoded, interleaved, and modulated Orthogonal Frequency Division Multiplexing signal that is used by access terminals operating within the coverage area of the access network. On the Forward Broadcast and Multicast Services Channel, the transmissions occur with single effective antenna at the access network. The access terminal may use receive diversity if it chooses to use multiple antennas.

The Forward Broadcast and Multicast Services Channel packet is generated by the BCMCS MAC Protocol 14.2.6, and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2.6.1. A CRC length of $N_{\text{CRC,DATA}}$ is used for this packet.

9.5.1.3.2.1 Forward Broadcast and Multicast Services Channel encoding

Each Forward Broadcast and Multicast Services Channel packet is generated by the BCMCS MAC Protocol 14.2.6.

9.5.1.3.2.2 Forward Broadcast and Multicast Services Channel data packet scrambling

A seed equal to $f_{\text{PHY-HASH}}(12 \times 512 \times 512 \times 16 + (m \bmod 512) \times 512 \times 16 + (l \bmod 512) \times 16 + (i \bmod 16))$ shall be used for the data scrambling operation. Here i is the superframe index in which the transmission of this packet started, l is the LOGIC_CHID and m denotes the SFN_ID.

9.5.1.3.2.2.1 Forward Broadcast and Multicast Services Channel data packet transmission

The data packet corresponding to the logical channel LOGIC_CHID shall be modulated on to the hop-ports assigned to this packet according to the following procedure:

- Initialize a port counter i , a HARQ retransmission counter r , a frame counter f , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- Let $F(r)$ be the total number of PHY frames to be used in the r^{th} HARQ retransmission of the packet, as specified by the 8.1.5. The frames shall be indexed $(r, 0), (r, 1) \dots (r, F(r) - 1)$.
- Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) . Let the resulting sequence be denoted by $p_0, p_1, \dots, p_{n-1}$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .
- Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} Orthogonal Frequency Division Multiplexing symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY frame (r, f) , which is a function of the packet format and HARQ retransmission index r . If n_{sc} is a Forward Data Channel available subcarrier, then a modulation symbol s from subpacket m with modulation order q is generated by the modulator according to the procedure described in 8.6, where m shall be equal to $(i_{\text{TILE}} N_{\text{SUBPACKETS-IN-TILE}} + (j N_{\text{BLOCK}} + i \bmod N_{\text{BLOCK}}) \bmod N_{\text{SUBPACKETS-IN-TILE}}) \bmod t$.

Here t is the total number of subpackets in the packet (equal to $N_{\text{DCH, SUBPACKETS}}$), N_{BLOCK} is the number of subcarriers in a block, $i_{\text{TILE}} = \lfloor i / N_{\text{BLOCK}} \rfloor$ and $N_{\text{SUBPACKETS-IN-TILE}}$ is computed as follows:

$$N_{\text{SUBPACKETS-IN-TILE}} = t \text{ if } i_{\text{TILE}} < (N_{\text{TILES}} \bmod t), \text{ where } N_{\text{TILES}} = \lfloor n / N_{\text{BLOCK}} \rfloor$$

$$N_{\text{SUBPACKETS-IN-TILE}} = \min \left(t, \left\lceil \frac{8t}{N_{\text{TILES}} - (N_{\text{TILES}} \bmod t)} \right\rceil \right) \text{ otherwise.}$$

- e) The modulation symbol s shall be modulated with power density P on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{P}s$. The modulation shall be done on the antenna with index 0. The same power density P shall be used over all the hop-ports assigned to this packet.
- f) Increment i . If $i = n$, increment j and set $i = 0$.
- g) If $j = N_{\text{FRAME}}$, set $j = 0$ and increment f .
- h) If $f = F(r)$, then increment r and set $f = 0$.
- i) If the last HARQ retransmission has been completed (as determined by the FTC MAC Protocol 8.6), then stop. Else repeat steps b) through h).

9.5.2 Supercast transmitter

This subclause defines requirements specific to access network equipment and operation for the support of supercast operation of unicast traffic on the broadcast portion.

The transmitter shall reside in each sector of the access network. These requirements apply to the transmitter in each sector.

Note that the support of supercast is optional both at the access terminal and the access network.

9.5.2.1 Frequency parameters

9.5.2.1.1 Channel spacing and designation

See 3GPP2 C.S0024 for a description of the band classes that an access network may support.

9.5.2.2 Modulation characteristics

Two radio configurations are defined for the Supercast services, which are described in 9.5.1.3.1.2.1 and 9.5.1.3.1.2.2 respectively.

9.5.2.2.1 Supercast signals

The Supercast channels are described in Table 225.

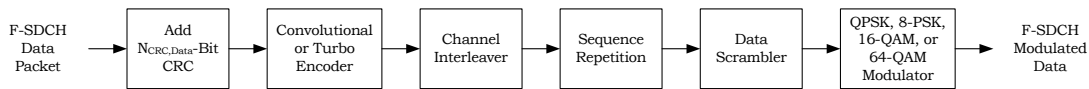
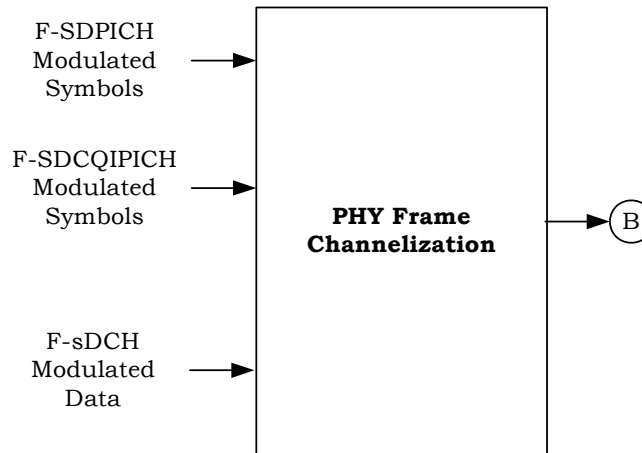
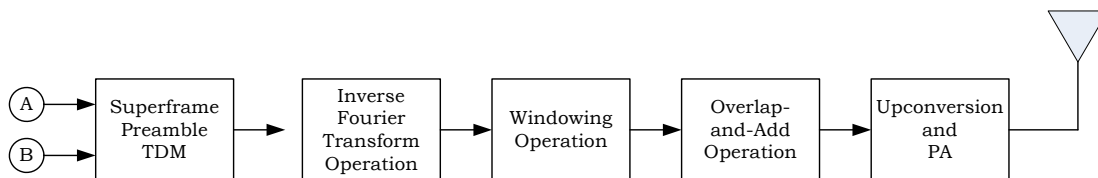
Table 225—Description of the supercast channels

F-SDPICH	Forward Superposed Dedicated Pilot Channel
F-SCQIPICH	Forward Superposed Channel Quality Indicator Pilot Channel
F-SDCH	Forward Superposed Data Channel

9.5.2.2.1.1 Channel structures

The Forward Supercast Channel consists of the channels specified in Table 225

The structure of the Forward Supercast Channel is shown in Figure 140. The channel structure for the single effective antenna case is shown in Figure 141 and Figure 142. The multiple antenna operation figures are similar to those described in Figure 127 through Figure 130.

**Figure 140—Forward superposed data channel structure****Figure 141—Channel structure of the PHY frames****Figure 142—Channel structure for the Single-Transmit-Antenna Case**

9.5.2.2.2 Forward Superposed Dedicated Pilot Channel

The Forward Superposed Dedicated Pilot Channel (F-SDPICH) is a pilot channel that helps in the channel estimation for the superposed traffic on the broadcast and multicast segment.

9.5.2.2.2.1 Structure for the Single-Transmit-Antenna Case Forward Superposed Dedicated Pilot Channel

The Forward Superposed Dedicated Pilot Channel is present in BRCH subzones. As described in 9.2.14.2.2, the BRCH subzone is divided into units of hop-port blocks. Each hop-port block consists of 16 contiguous hop-ports, which are mapped by the hopping permutation to a contiguous set of subcarriers. Also, the set of subcarriers corresponding to a hop-port block does not change over one PHY frame. (Note however that since the Forward Link supports SDMA, a two hop-port blocks can be mapped to the same set of subcarriers.) Therefore, the set of resources (over time and frequency) in a BRCH subzone can be divided into units of tiles, where a tile is a contiguous 16x7 rectangle of hop-ports (16 in frequency and 7 in time), which are mapped to a contiguous 16x7 rectangle of subcarriers (16 in frequency and 7 in time).

Each tile in a BRCH subzone can be assigned to the control segment, to the Forward Superposed Data Channel, or can be left blank. The Forward Superposed Dedicated Pilot Channel shall be present in each tile in a BRCH subzone, i.e., some of the subcarriers in each tile shall be designated as the Forward Superposed Dedicated Pilot Channel subcarriers. Each tile in a BRCH subzone may be transmitted from up to four tile-antennas, where a tile-antenna is as defined in Clause 3. The Forward Superposed Dedicated Pilot Channel waveform shall be defined separately from each of these tile-antennas. The tile-antennas used to transmit the Forward Superposed Dedicated Pilot Channel in a tile shall be the same as the tile-antennas used to transmit the control segment or the Forward Superposed Data Channel from that tile. If two tiles map to the same frequency resources, then the Forward Superposed Dedicated Pilot Channel waveforms assigned to these tiles shall be superimposed. The Forward Superposed Dedicated Pilot Channel configuration in each tile depends on the following parameters:

- The number of tile-antennas n_t : n_t is equal to 1 if the tile is occupied by the control segment. If the tile is occupied by the Forward Superposed Data Channel, the value of n_t is the same as the number of tile-antennas used to transmit the Forward Superposed Data Channel from that tile.
- Forward Superposed Dedicated Pilot Channel format: The Forward Superposed Dedicated Pilot Channel format can take one of two values, 0 and 1. The Forward Superposed Dedicated Pilot Channel format 0 shall be used for tiles occupied by the control segment. For tiles occupied by the Forward Superposed Data Channel, the Forward Superposed Dedicated Pilot Channel format depends on the Forward Superposed Data Channel assignment occupying this tile, and is determined by the FTC MAC Protocol.
- Energy per modulation symbol: This quantity, denoted by P , is defined separately for each tile-antenna and each tile, but is fixed for all the modulation symbols from the same tile-antenna within a tile. For tiles in the control segment, the exact procedure determination of P is outside the scope of this standard. For tiles that are occupied by the Forward Superposed Data Channel, the energy per modulation symbol from a given tile-antenna is the same as the energy per modulation symbol used to transmit the Forward Superposed Data Channel from that tile-antenna in that tile.
- CodeOffset: This is an integer between 0 and 3. It takes value 0 for tiles belonging to the control segment. For tiles belonging to the Forward Superposed Data Channel, the value is determined by the value of SubtreeIndex for that Forward Superposed Data Channel assignment, which is determined by the FTC MAC Protocol. For each value of SubtreeIndex, the value of CodeOffset is given by $\text{FLDPICHCodeOffset}_{\text{SubtreeIndex}}$, which is a field of the Overhead Messages Protocol.

The locations of the Forward Superposed Dedicated Pilot Channel subcarriers in a tile depend on the Forward Superposed Dedicated Pilot Channel format and are shown in Figure 143. Note that the hop-ports within a tile are indexed 0 to 15 in increasing order of hop-port index, and the Orthogonal Frequency Division Multiplexing symbols within a Forward Link PHY frame are indexed 0 to 6 with the earliest Orthogonal Frequency Division Multiplexing symbol being indexed 0.

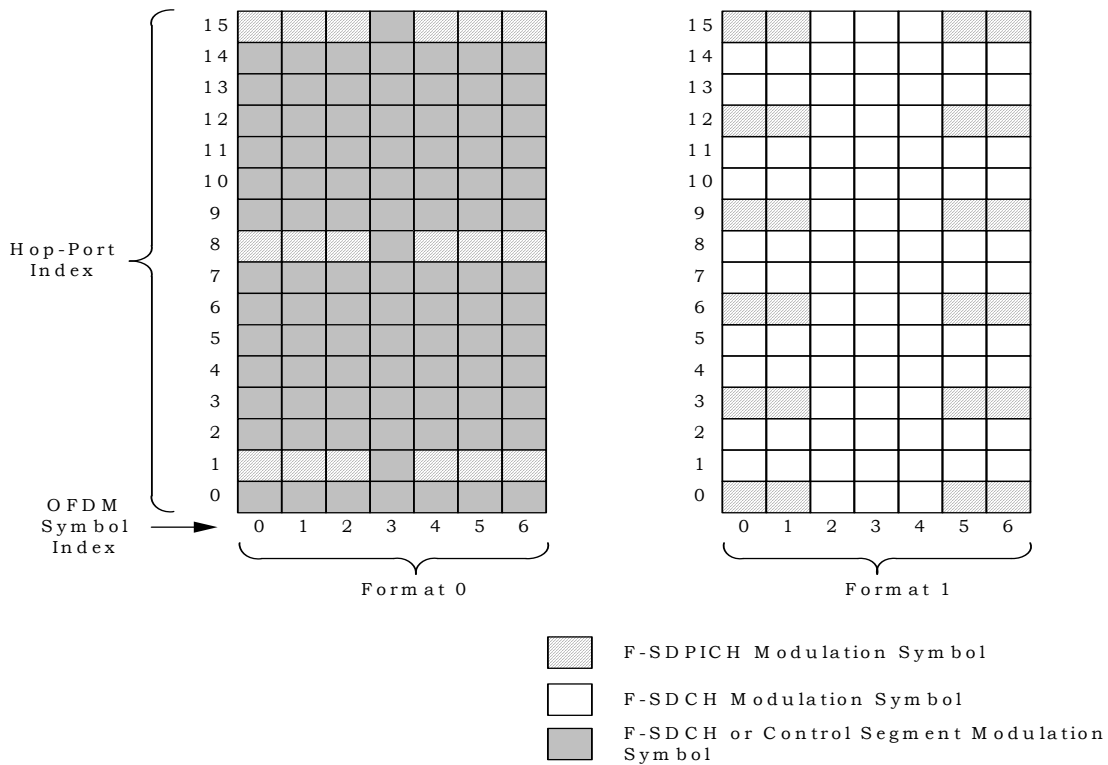


Figure 143—Location of Forward Superposed Dedicated Pilot Channel Subcarriers within a tile for the Different Forward Superposed Dedicated Pilot Channel formats

9.5.2.2.2.1.1 Forward Superposed Dedicated Pilot Channel Format 0

For the Forward Superposed Dedicated Pilot Channel Format 0, the Forward Superposed Dedicated Pilot Channel shall occupy the modulation symbol of the tile if the hop-port index within the tile is in the set {1, 8, 15} and the Orthogonal Frequency Division Multiplexing symbol index t within the Forward Link PHY frame is in the set $T = \{0, 1, 2, 4, 5, 6\}$, provided none of those symbols is a BeaconOnlyOFDMSymbol.

The complex value of the Forward Superposed Dedicated Pilot Channel modulation symbol on the tile-antenna with index k shall depend only on the Orthogonal Frequency Division Multiplexing symbol index t and shall be given by

$$S_{t,k} = \sqrt{P} \exp \left(\frac{j2\pi}{3} (k + \text{CodeOffset}) F_{\text{PHASE}}(t) \right)$$

where j denotes the complex number (0, 1), and P denotes the energy per modulation symbol on tile-antenna k used by the Forward Superposed Dedicated Pilot Channel. The function $F_{\text{PHASE}}(t)$ maps the set T to $\{0, 1, 2, 2, 1, 0\}$. Thus, $F_{\text{PHASE}}(0) = 0$, $F_{\text{PHASE}}(1) = 1$, $F_{\text{PHASE}}(2) = 2$, $F_{\text{PHASE}}(4) = 2$, $F_{\text{PHASE}}(5) = 1$, $F_{\text{PHASE}}(6) = 0$.

9.5.2.2.2.1.2 Forward Superposed Dedicated Pilot Channel Format 1

For the Forward Dedicated Pilot Channel Format 1, the Forward Superposed Dedicated Pilot Channel shall occupy the modulation symbol of the tile if the hop-port index within the tile is in the set $\{0, 3, 6, 9, 12, 15\}$

and the Orthogonal Frequency Division Multiplexing symbol index, t , is in the set $T = \{0, 1, 5, 6\}$, when none of those symbols is a BeaconOnlyOFDMSymbol.

The complex value of the Forward Superposed Dedicated Pilot Channel modulation symbol on the tile antenna with index k shall be given by

$$S_{t,k} = \sqrt{P} \exp(j\pi(k + \text{CodeOffset})F_{\text{PHASE}}(t))$$

where j denotes the complex number (0, 1), and P denotes the energy per modulation symbol on tile-antenna k used by the Forward Dedicated Pilot Channel. The function $F_{\text{PHASE}}(t)$ maps the set T to $\{0, 1, 1, 0\}$. Thus, $F_{\text{PHASE}}(0) = 0$, $F_{\text{PHASE}}(1) = 1$, $F_{\text{PHASE}}(5) = 1$, $F_{\text{PHASE}}(6) = 0$.

9.5.2.2.2.1.3 Forward Superposed Dedicated Pilot Channel scrambling

9.5.2.2.2.1.3.1 Forward Superposed Dedicated Pilot Channel index definition

Forward Superposed Dedicated Pilot Channel scrambling is done on a tile-by-tile basis for the Forward Superposed Dedicated Pilot Channel. The scrambling symbols that shall be used shall be those generated for subcarriers that correspond to the Forward Superposed Dedicated Pilot Channel hop-ports (via the hop-permutation), as defined in 9.2.14.4. These subcarriers are henceforth referred to as the Forward Superposed Dedicated Pilot Channel subcarriers. For the purpose of scrambling, the Forward Superposed Dedicated Pilot Channel subcarriers in each tile or quarter-tile shall be indexed by a quantity called the Forward Superposed Dedicated Pilot Channel index. The Forward Superposed Dedicated Pilot Channel index shall be computed according to the following procedure for the Forward Superposed Data Channel tiles:

- a) Initialize an Orthogonal Frequency Division Multiplexing symbol counter i , a subcarrier counter j and a Forward Superposed Dedicated Pilot Channel index counter k to 0.
- b) If the subcarrier j in Orthogonal Frequency Division Multiplexing symbol i within the tile is a Forward Superposed Dedicated Pilot Channel subcarrier, then
 - 1) Set its Forward Superposed Dedicated Pilot Channel index to k .
 - 2) Increment k by 1.
- c) Increment i by 1. If $i = N_{\text{FRAME}}$, set i to 0 and increment j .
- d) Repeat steps b) and c) until $j = N_{\text{BLOCK}}$.

In other words, the Forward Superposed Dedicated Pilot Channel subcarriers are indexed in time first, followed by frequency.

9.5.2.2.2.1.3.2 Scrambling sequence

The scrambling symbols for a tile depend on the tile index T , which shall be equal to $(f_{\text{MIN}} - N_{\text{GUARD, LEFT}}) / N_{\text{BLOCK}}$, where f_{MIN} is the lowest indexed subcarrier in that tile. Let $\{t_2 t_1 t_0\}$ be the 3 LSBs of T . Let $b_8, b_7, \dots, b_0$ be the 9 bits of PilotPhase, with b_8 being the MSB and b_0 being the LSB. For the tile with index T within any PHY frame in the superframe with index SFInd , a complex scrambling sequence shall be generated using the common complex scrambling algorithm described in 9.2.5.2 with seed $[01111001 t_2 t_1 t_0 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0]$. The k^{th} symbol $c(k)$ in the complex scrambling sequence shall be used to scramble the Forward Superposed Dedicated Pilot Channel subcarrier with Forward Superposed Dedicated Pilot Channel index k . The scrambling operation shall consist of multiplying the unscrambled complex symbol on the subcarrier with the scrambling symbol $c(k)$.

9.5.2.2.3 Forward Superposed Channel Quality Indicator Pilot Channel

The Forward Superposed Channel Quality Indicator Pilot Channel (F-SCQIPICH) is a pilot channel that helps in the channel quality feedback for the superposed traffic on the broadcast and multicast section.

9.5.2.2.3.1 Forward Superposed Channel Quality Indicator Pilot Channel structure

The Forward Superposed Channel Quality Indicator Pilot Channel shall be present in Forward Link PHY frames satisfying $j \bmod \text{NumFLInterlaces} = 4$ if $\text{DuplexingMode} = \text{FDD}$ and in Forward Link PHY frames satisfying $j \bmod \text{NumFLInterlaces} = 0$ if $\text{DuplexingMode} = \text{TDD44}$ or TDD63 . Here j denotes the index of the Forward Link PHY frame in the superframe. In the TDD mode, the Forward Superposed Channel Quality Indicator Pilot Channel shall be present in Forward Link PHY frames satisfying $j \bmod 4 = 2$, j denotes the index of the Forward Link PHY frame in the superframe. In these Forward Link PHY frames, the Forward Superposed Channel Quality Indicator Pilot Channel shall be present on the Orthogonal Frequency Division Multiplexing symbols with indices 3 and 4 in the Forward Link PHY frame, where the Orthogonal Frequency Division Multiplexing symbols in the Forward Link PHY frame are indexed from 0 to 6. The Forward Superposed Channel Quality Indicator Pilot Channel is designed so as to enable the access terminal to estimate channel quality for reporting the $r\text{-cqich}$, and to estimate the optimal precoding matrix for reporting the $r\text{-bfch}$. The notion of a precoding matrix is defined in 9.2.8.2.

The Forward Superposed Channel Quality Indicator Pilot Channel is transmitted on a disjoint set of subcarriers from each effective antenna, with the number of transmit antennas being given by the $\text{NumEffectiveAntennas}$ field of the Overhead Messages Protocol. Each subcarrier occupied by the Forward Superposed Channel Quality Indicator Pilot Channel from a given transmit antenna shall be modulated with the value $(\sqrt{P}, 0)$ from that effective antenna, where P is given by the $\text{CQIPilotTransmitPower}$ field of the Overhead Messages Protocol. The remaining effective antennas shall be left unmodulated on this subcarrier.

For the Orthogonal Frequency Division Multiplexing symbol with index 3 within a Forward Link PHY frame containing the Forward Superposed Channel Quality Indicator Pilot Channel, a usable subcarrier with index i_{sc} shall be modulated with the Forward Superposed Channel Quality Indicator Pilot Channel from the antenna with index k if $i_{sc} \bmod 16 = a_k$. For the Orthogonal Frequency Division Multiplexing symbol with index 4 within a Forward Link PHY frame containing the Forward Superposed Channel Quality Indicator Pilot Channel, a usable subcarrier with index i_{sc} shall be modulated with the Forward Channel Quality Indicator Pilot Channel from the antenna with index k if $i_{sc} \bmod 16 = b_k$, where, a_k and b_k are as shown in Table 226, and have been chosen so as to ensure that the Forward Superposed Channel Quality Indicator Pilot Channel does not collide with the Forward Superposed Dedicated Pilot Channel.

Table 226—Values of the parameters a_k and b_k

Antenna index (k)	a_k	b_k
0	2	10
1	3	11
2	4	12
3	5	13
4	10	2
5	11	3
6	12	4
7	13	5

9.5.2.2.3.2 Forward Superposed Channel Quality Indicator Pilot Channel scrambling

For the purpose of scrambling, the Forward Superposed Channel Quality Indicator Pilot Channel subcarriers in frame on each effective antenna shall be indexed by a quantity called the Forward Superposed Channel

Quality Indicator Pilot Channel index. The Forward Superposed Channel Quality Indicator Pilot Channel index of a subcarrier on effective antenna k shall be computed according to the following procedure:

- a) Initialize an Orthogonal Frequency Division Multiplexing symbol counter i to 3, a subcarrier counter j and a Forward Superposed Channel Quality Indicator Pilot Channel index counter r to 0.
- b) If the subcarrier j in Orthogonal Frequency Division Multiplexing symbol i is a Forward Superposed Channel Quality Indicator Pilot Channel subcarrier, then
 - 1) Set its Forward Superposed Channel Quality Indicator Pilot Channel index on antenna k to r .
 - 2) Increment r by 1.
- c) Increment j by 1. If $j = N_{\text{FFT}}$, set j to 0 and increment i by 1.
- d) Repeat steps b) and c) until $i = 5$.

In other words, the Forward Superposed Channel Quality Indicator Pilot Channel subcarriers are indexed in frequency first, followed by time.

A complex scrambling sequence shall be generated using the common complex scrambling algorithm described in 9.2.5.2 with seed $[0101k_2k_1k_0f_3f_2f_1f_0b_8b_7b_6b_5b_4b_3b_2b_1b_0]$. Here $[k_2k_1k_0]$ are the three LSBs of the antenna index k , $(f_3f_2f_1f_0)$ are the four LSBs of the superframe index and $[b_8b_7b_6b_5b_4b_3b_2b_1b_0]$ is equal to the PilotPN. The r^{th} symbol $c(r)$ in the complex scrambling sequence shall be used to scramble the Forward Superposed Channel Quality Indicator Pilot Channel subcarrier with the Forward Superposed Channel Quality Indicator Pilot Channel index r on effective antenna k . The scrambling operation shall consist of multiplying the unscrambled complex symbol on the subcarrier with the scrambling symbol $c(r)$.

9.5.2.2.4 Forward Superposed Data Channel

The Forward Superposed Data Channel (F-SDCH) consists of one or more data packets, which can span one Forward Link Frame. The Forward Link FHY Frame on which the packets are transmitted is determined by the FTC MAC Protocol Superposed. Each data packet is also assigned a set of hop-ports in each PHY frame of transmission by the FTC MAC Protocol Superposed. Each data packet is further associated with a packet format index, which is also assigned by the FTC MAC Protocol (see 8.6).

9.5.2.2.4.1 Forward Superposed Data Channel available subcarriers

Not all subcarriers may be available for modulation by the Forward Superposed Data Channel. For example, subcarriers in which the Forward Superposed Dedicated Pilot Channel is transmitted cannot be used by the Forward Superposed Data Channel. In this subclause, the notion of Forward Superposed Data Channel unavailable subcarriers is defined. A subcarrier is unavailable for the Forward Superposed Data Channel if:

- The subcarrier is a pilot subcarrier i.e., it is allocated to one of the Forward Link Pilot Channels (the Forward Superposed Dedicated Pilot Channel and the Forward Superposed Channel Quality Indicator Pilot Channel).
- The subcarrier is part of a BeaconOnlyOFDMsymbol.

All subcarriers that are not unavailable as defined above shall be referred to as “Forward Superposed Data Channel available subcarriers.”

9.5.2.2.4.2 Forward Superposed Data Channel Single Input Single Output Mode

The Forward Superposed Data Channel consists of a data packet, which can span one Forward Link PHY frame. The Forward Link PHY frame on which this packet is transmitted is determined by the Lower MAC Layer (see Clause 8). Each data packet and erasure sequence is also assigned a set of hop-ports in the PHY frame of transmission by the Lower MAC Layer (see Clause 8). Each data packet is further associated with a packet format index, which is also assigned by the Lower MAC Layer (see Clause 8).

9.5.2.2.4.2.1 Forward Superposed Data Channel Packet Encoding for Single Input Single Output

The Forward Superposed Data Channel packet is generated by the FTC MAC Protocol (see 8.6), and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2.6.1. A CRC length of $N_{\text{CRC,Data}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(12 \times 512 \times 512 \times 16 + (m \bmod 512) \times 512 \times 16 + (p \bmod 512) \times 16 + (i \bmod 16))$ shall be used for the data scrambling operation, where i is the superframe index in which the transmission of this packet started and p is the PilotPN. m denotes the MACID of the access terminal of interest except in the case of multicast group resource transmissions. For the case of multicast group resource transmissions, m denotes the GroupID.

9.5.2.2.4.2.2 Forward Superposed Data Channel Data Packet Transmission for Single Input Single Output

The data packet shall be modulated on to the hop-ports assigned to this packet according to the following procedure. Although the procedure is specified for a generic $F(r)$, $F(r) = 1$ for the Forward Superposed Data Channel.

- Initialize a port counter i , a HARQ retransmission counter r , a frame counter f , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- Let $F(r)$ be the total number of PHY frames to be used in the r^{th} HARQ retransmission of the packet, as specified by the 8.6. The frames shall be indexed $(r, 0), (r, 1) \dots (r, F(r) - 1)$.
- Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) . Let the resulting sequence be denoted by $p_0, p_1, \dots, p_{n-1}$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .
- Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} Orthogonal Frequency Division Multiplexing symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY frame (r, f) , which is a function of the packet format and HARQ retransmission index r . If n_{sc} is a Forward Superposed Data Channel available subcarrier, then a modulation symbol s from subpacket m with modulation order q is generated by the modulator according to the procedure described in 9.2.7.3, where m shall be equal to $(i_{\text{TILE}} + (j + i \bmod N_{\text{BLOCK}}) \bmod N_{\text{SUBPACKETS-IN-TILE}}) \bmod t$. Here t is the total number of subpackets in the packet (equal to $N_{\text{DCH, SUBPACKETS}}$), N_{BLOCK} is the number of subcarriers in a block, $i_{\text{TILE}} = \lfloor i / N_{\text{BLOCK}} \rfloor$ and $N_{\text{SUBPACKETS-IN-TILE}}$ is computed as follows:

$$N_{\text{SUBPACKETS-IN-TILE}} = t \text{ if } i_{\text{TILE}} < (N_{\text{TILES}} \bmod t), \text{ where } N_{\text{TILES}} = \lfloor n / N_{\text{BLOCK}} \rfloor$$

$$N_{\text{SUBPACKETS-IN-TILE}} = \min \left(t, \left\lceil \frac{8t}{N_{\text{TILES}} - (N_{\text{TILES}} \bmod t)} \right\rceil \right) \text{ otherwise.}$$

- The modulation symbol s shall be modulated with power density P on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{P}s$. The modulation shall be done on the tile-antenna with index 0 since i_{TILE} is a BRCH resource.
- Different values of power density P may be used for different BRCH resources. Determining the value of P is out of the scope of this standard.
- Increment i . If $i = n$, increment j and set $i = 0$.
- If $j = N_{\text{FRAME}}$, set $j = 0$ and increment f .
- If $f = F(r)$, then increment r and set $f = 0$.
- If the last HARQ retransmission has been completed [as determined by the FTC MAC Protocol (see 8.6)], then stop. Else repeat steps c) through h).

9.5.2.2.4.3 Forward Superposed Data Channel Precoding for Multiple Input Multiple Output

If precoding is used on the Forward Superposed Data Channel, the tile antennas used for Multiple Input Multiple Output or Space Time Transmit Diversity transmissions are obtained from the effective antennas through the use of precoding matrices as described in 9.2.8.2. When precoding is used by the access network, these tile antennas shall be used for the Space Time Transmit Diversity, Multi-Codeword and Single Codeword modes as described in 9.4.1.2.5.6, 9.4.1.2.5.7, and 9.4.1.2.5.8.

In the BRCH mode, the only mode available for the Forward Superposed Data Channel, the access network can choose to use any precoding matrix.

9.5.2.2.4.4 Forward Superposed Data Channel Multiple Input Multiple Output Multi-Codeword Mode

Multiple data packets may be transmitted in Multiple Input Multiple Output Multi-Codeword mode. The number of packets is equal to N_{LAYERS} , the number of layers for this transmission as specified by the FTC MAC Protocol (see 8.6). The layers shall be indexed 0 through $N_{\text{LAYERS}} - 1$. A separate packet shall be transmitted on each layer.

9.5.2.2.4.4.1 Forward Superposed Data Channel permutation matrices

Let $P_p^{\text{NUM\_LAYER}}$ denote the set of all permutation matrices of order NUM_LAYER ($p = 0, 1, \dots, \text{NUM\_LAYER}! - 1$). The set of all such matrices for NUM_LAYER = 1, 2, 3, and 4 are enumerated in 9.2.8.3.

9.5.2.2.4.4.2 Forward Superposed Data Channel Data Packet Encoding for Multi-Codeword Multiple Input Multiple Output

Each Forward Superposed Data Channel packet is generated by the FTC MAC Protocol (see 8.6), and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2.6. A CRC length of $N_{\text{CRC,Data}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(13 \times 512 \times 512 \times 16 + (m \bmod 512) \times 512 \times 16 + (p \bmod 512) \times 16 + (i \bmod 16))$ shall be used for the data scrambling operation, where i is the superframe index in which the transmission of this packet started, p is the PilotPN and m is the MACID of the access terminal of interest.

9.5.2.2.4.4.3 Forward Superposed Data Channel Data Packet Transmission for Multi-Codeword Multiple Input Multiple Output

The N_{LAYERS} data packets shall be modulated on to the hop-ports assigned to this packet according to the following procedure. Although the procedure is specified for a generic $F(r)$, $F(r)=1$ for the Forward Superposed Data Channel.

- Initialize a port counter i , a HARQ retransmission counter r , a frame counter f , a permutation counter p , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- Let $F(r)$ be the total number of PHY frames to be used by the r^{th} HARQ retransmission of the packet, as specified by the 8.6. The frames shall be indexed $(r, 0), (r, 1) \dots (r, F(r) - 1)$.
- Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) . Let the resulting sequence be denoted by $p_0, p_1, \dots, p_{n-1}$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .
- Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} Orthogonal Frequency Division Multiplexing symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY frame (r, f) , which is a function of the packet format and HARQ retransmission index r . If n_{sc} is not a pilot subcarrier and is a Forward Superposed Data Channel available subcarrier, then a sequence of N_{LAYERS} modulation symbols $\{s_0, s_1, \dots, s_{N_{\text{LAYERS}}-1}\}$ from subpackets $\{m_0, m_1, \dots, m_{N_{\text{LAYERS}}-1}\}$

1} respectively with modulation order q is generated by the modulator according to the procedure described in 9.2.6. The subpacket m_k of the data packet on layer k shall be equal to $(i_{\text{TILE}} + (j + i \bmod N_{\text{BLOCK}}) \bmod N_{\text{SUBPACKETS-IN-TILE}}) \bmod t$. Here t is the total number of subpackets in the packet (equal to $N_{\text{DCH, SUBPACKETS}}$ for that layer), N_{BLOCK} is the number of subcarriers in a block, $i_{\text{TILE}} = \lfloor i / N_{\text{BLOCK}} \rfloor$ and $N_{\text{SUBPACKETS-IN-TILE}}$ is computed as follows:

$$N_{\text{SUBPACKETS-IN-TILE}} = t \text{ if } i_{\text{TILE}} < (N_{\text{TILES}} \bmod t), \text{ where } N_{\text{TILES}} = \lfloor n / N_{\text{BLOCK}} \rfloor$$

$$N_{\text{SUBPACKETS-IN-TILE}} = \min \left(t, \left\lceil \frac{8t}{N_{\text{TILES}} - (N_{\text{TILES}} \bmod t)} \right\rceil \right) \text{ otherwise.}$$

- e) This modulation symbol s_k ($k = 0, 1, \dots, N_{\text{LAYERS}} - 1$) shall be modulated with power density P on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{P}s_k$. Different values of power density P may be used for different BRCH resources.
- f) Define $y_k = \sqrt{P}s_k$. Let $\mathbf{Y}$ be the vector $\{y_k\}$, $k = 0, 1, \dots, N_{\text{LAYERS}} - 1$. Define $z_k = (P_{\text{P}}^{\text{NUM\_LAYER}} \mathbf{Y})_k$.
- g) z_k shall be transmitted on the antenna with index k .
- h) Increment i . If $i = n$, increment j and set $i = 0$.
- i) Increment p . If $p = (N_{\text{LAYERS}})!$, set $p = 0$.
- j) If $j = N_{\text{FRAME}}$, set $j = 0$ and increment f .
- k) If $f = F(r)$, then increment r and set $f = 0$.
- l) If the last HARQ retransmission has been completed [as determined by the FTC MAC Protocol (see 8.6)], then stop. Else, repeat steps b) through h).

9.5.2.2.4.5 Forward Superposed Data Channel Multiple Input Multiple Output Single Codeword mode

A single data packet shall be transmitted in Multiple Input Multiple Output Single Codeword mode.

9.5.2.2.4.5.1 Forward Superposed Data Channel Data Packet Encoding for Multiple Input Multiple Output Multi-Codeword

The Forward Superposed Data Channel packet is generated by the FTC MAC Protocol (see 8.6), and is split, appended with CRC, encoded, channel interleaved, repeated, data-scrambled and modulated according to the procedure described in 9.2.6. A CRC length of $N_{\text{CRC,Data}}$ is used for this packet. A seed equal to $f_{\text{PHY-HASH}}(14 \times 512 \times 512 \times 16 + (m \bmod 512) \times 512 \times 16 + (p \bmod 512) \times 16 + (i \bmod 16))$ shall be used for the data scrambling operation, where i is the superframe index in which the transmission of this packet started, p is the PilotPN, and m is the MACID of the access terminal of interest.

9.5.2.2.4.5.2 Forward Superposed Data Channel Data Packet Transmission for Multiple Input Multiple Output Multi-Codeword

The data packet shall be modulated on to the hop-ports assigned to this packet according to the following procedure. Although the procedure is specified for a generic $F(r)$, $F(r)=1$ for the Forward Superposed Data Channel.

- a) Initialize a port counter i , a HARQ retransmission counter r , a frame counter f , and an Orthogonal Frequency Division Multiplexing symbol counter j all to 0.
- b) Let $F(r)$ be the total number of PHY frames to be used by the r^{th} HARQ retransmission of the packet, as specified by the 8.6. The frames shall be indexed $(r, 0), (r, 1) \dots (r, F(r) - 1)$.
- c) Arrange the set of usable hop-ports assigned to this packet in PHY frame (r, f) . Let the resulting sequence be denoted by $p_0, p_1, \dots, p_{n-1}$, where n is the total number of usable hop-ports assigned to this packet in PHY frame (r, f) .

- d) Let n_{sc} be the subcarrier index corresponding to the hop-port p_i in the j^{th} Orthogonal Frequency Division Multiplexing symbol in PHY frame (r, f) . Let q be the modulation order to be used in PHY frame (r, f) , which is a function of the packet format and HARQ retransmission index r . If n_{sc} is a Forward Superposed Data Channel available subcarrier, then a sequence of N_{LAYERS} modulation symbols $\{s_0, s_1, \dots, s_{N_{\text{LAYERS}}-1}\}$ from subpacket m with modulation order q is generated by the modulator according to the procedure described in 9.2.6, where m is equal to $(i_{\text{TILE}} + (j + i \bmod N_{\text{BLOCK}}) \bmod N_{\text{SUBPACKETS-IN-TILE}}) \bmod t$, t is the total number of subpackets in the packet (equal to $N_{\text{DCH, SUBPACKETS}}$), N_{BLOCK} is the number of subcarriers in a block, $i_{\text{TILE}} = \lfloor i/N_{\text{BLOCK}} \rfloor$ and $N_{\text{SUBPACKETS-IN-TILE}}$ is computed as follows:

$$N_{\text{SUBPACKETS-IN-TILE}} = t \text{ if } i_{\text{TILE}} < (N_{\text{TILES}} \bmod t), \text{ where } N_{\text{TILES}} = \lfloor n/N_{\text{BLOCK}} \rfloor$$

$$N_{\text{SUBPACKETS-IN-TILE}} = \min \left(t, \left\lceil \frac{8t}{N_{\text{TILES}} - (N_{\text{TILES}} \bmod t)} \right\rceil \right) \text{ otherwise.}$$

- e) This modulation symbol s_k ($k = 0, 1, \dots, N_{\text{LAYERS}} - 1$) shall be modulated with power density P on hop-port p_i , i.e., the value of the corresponding subcarrier shall be $\sqrt{P}s_k$. The modulation shall be done on the antenna with index k if i_{TILE} is a DRCH resource, and on the tile-antenna with index k since i_{TILE} is a BRCH resource.
- f) Different values of power density P may be used for different BRCH resources. Determining the value of P is out of the scope of this standard.
- g) Increment i . If $i = n$, increment j and set $i = 0$.
- h) If $j = N_{\text{FRAME}}$, set $j = 0$ and increment f .
- i) If $f = F(r)$, then increment r and set $f = 0$.
- j) If the last HARQ transmission has been completed [as determined by the FTC MAC Protocol (see 8.6)], then stop. Else, repeat steps b) through h).

9.5.3 Receiver

9.5.3.1 Channel spacing and designation

Channel spacing and designations for the access network reception shall be as specified in 9.3.1.1.1.

9.5.3.2 Demodulation characteristics

The access network demodulation process shall perform complementary operations to the access terminal modulation process.

List of Physical and Logical Channels for the Wideband mode

- Forward Acknowledgment Channel
- Forward Acquisition Channel
- Forward Beacon Pilot Channel
- Forward Broadcast and Multicast Pilot Channel
- Forward Broadcast and Multicast Services Channel
- Forward Cell Null Channel
- Forward Channel Quality Indicator Pilot Channel
- Forward Common Pilot Channel
- Forward Data Channel
- Forward Dedicated Pilot Channel

- Forward Fast Other Sector Interference Channel
- Forward Interference over Thermal Channel
- Forward Other Sector Interference Channel
- Forward Pilot Quality Indicator Channel
- Forward Power Control Channel
- Forward Preamble Pilot Channel
- Forward Primary Broadcast Channel
- Forward Quick Paging Channel
- Forward Reverse Activity Bit Channel
- Forward Secondary Broadcast Channel
- Forward Shared Control Channel
- Forward Start-of-Packet Channel
- Reverse Access Channel
- Reverse Acknowledgment Channel.
- Reverse Auxiliary Pilot Channel
- Reverse Code Division Multiple Access Control Channel
- Reverse Code Division Multiple Access Data Channel
- Reverse Dedicated Pilot Channel.
- Reverse Code Division Multiple Access Dedicated Control Channel (R-CDCCH)
 - r-cqich (channel quality indicator channel)
 - r-reqch (request channel)
 - r-pahch (power amplifier headroom channel)
 - r-psdch (spectral density indication channel)
- Reverse Orthogonal Frequency Division Multiple Access Data Channel
- Reverse Orthogonal Frequency Division Multiple Access Control Channel
 - r-mqich
 - r-cqich
 - r-reqch
 - r-sfch
 - r-bfch
- Reverse Pilot Channel
- Reverse Traffic Channel

10. Security functions

10.1 Introduction

Security Function consists of following protocols:

- AES Ciphering Protocol
- Basic Message Integrity Protocol
- Basic Key Exchange Protocol

10.2 AES Ciphering Protocol

The AES Ciphering Protocol uses the AES (a.k.a. Rijndael) procedures defined in 3GPP2 S.S0055Av2.0 in order to encrypt and decrypt the Radio Link Protocol packets.

10.2.1 Primitives and public data

10.2.1.1 Commands

This protocol does not define any commands.

10.2.1.2 Return indications

This protocol does not return any indications.

10.2.1.3 Procedure calls

- ENCRYPT
 - Inputs: Direction, DataUnit, StreamID, RouteCounter, SARResetCounter, SARSequenceNumber, SARSequenceRolloverCounter, PayloadSize, Payload
 - Outputs: Payload (Encrypted), CipheringKeyIndex

Possible values of each input and output are as follows:

- *Direction* – '0' (ForwardLink) or '1' (ReverseLink)
- *DataUnit* – '00' (Octets), '01' (RLP Packet Payload)
- *StreamID* – 16-bit hexadecimal number
- *RouteCounter* – 15-bit hexadecimal number
- *SARResetCounter* – 8-bit hexadecimal number
- *SARSequenceNumber* – Hexadecimal number of n bits, where n is less than or equal to 48.
- *SARSequenceRolloverCounter* – Hexadecimal number of (48 – length of *SARSequenceNumber*) bits.
- *PayloadSize* – Payload size in octets
- *Payload* – Payload to be encrypted
- *Payload* (Encrypted) – Encrypted payload of same size as input *Payload* and is available at same memory space as the input *Payload*.
- *CipheringKeyIndex* – '00' (not encrypted); '01', '10', '11' (encrypted with key corresponding to index '01', '10', or '11' respectively)
- DECRYPT

- Inputs: Direction, DataUnit, StreamID, RouteCounter, SARResetCounter, SARSequenceNumber, SARSequenceRolloverCounter, PayloadSize, Payload, CipherringKeyIndex
- Outputs: Payload (Decrypted)

Possible values of each input and output are as follows:

- *Direction* – '0' (ForwardLink) or '1' (ReverseLink)
- *DataUnit* – '00' (Octets), '01' (RLP Packet Payload)
- *StreamID* – 16-bit hexadecimal number
- *RouteCounter* – 15-bit hexadecimal number
- *SARResetCounter* – 8-bit hexadecimal number
- *SARSequenceNumber* – Hexadecimal number of n bits, where n is less than or equal to 48.
- *SARSequenceRolloverCounter* – Hexadecimal number of (48 – length of *SARSequenceNumber*) bits.
- *PayloadSize* – Payload size in octets
- *Payload* – Payload to be decrypted
- *CipherringKeyIndex* – '00' (not encrypted); '01', '10', '11' (encrypted with key corresponding to index '01', '10', or '11' respectively)
- *Payload* (Decrypted) – Decrypted payload of same size as input *Payload* and is available at same memory space as the input *Payload*.

10.2.1.4 Local common data

This protocol does not define any local common data.

10.2.1.5 Public data

This protocol shall make the following data public:

- Subtype for this protocol
- All data defined as Static Attribute, Static Non-Attribute Data, and local common data

10.2.2 Protocol data unit

This protocol does not transmit or receive data. This protocol provides encryption and decryption services to Radio Link Protocol.

10.2.3 Procedures and messages for the InConfiguration instance of the protocol

10.2.3.1 Protocol Initialization for the InConfiguration Protocol Instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

10.2.3.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values.

10.2.3.3 Message formats

This protocol does not define any messages.

10.2.4 Procedures and messages for the InUse instance of the protocol

10.2.4.1 Procedures

10.2.4.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in 15.1.2.

10.2.4.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

10.2.4.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

10.2.4.1.4 Constructing the Ciphering Key

The AES Ciphering Protocol shall construct the Ciphering keys as follows:

- If the value of the FTCCiphering attribute is equal to 0x01, then the protocol shall construct the Ciphering key for the Forward Traffic Channel, FTCCipheringKey, as follows:
 - If the FACCipheringKey[KeyIndex] public data of the Key Exchange Protocol is set to NULL, the protocol shall set FTCCipheringKey to NULL.
 - Otherwise, the protocol shall perform the following:
 - If the length of FACCipheringKey[KeyIndex] is equal to 128, then FTCCipheringKey shall be set to FACCipheringKey[KeyIndex].
 - Otherwise, if the length of FACCipheringKey[KeyIndex] is greater than 128, then FTCCipheringKey shall be the 128 most significant bits of FACCipheringKey[KeyIndex].
 - Otherwise, if the length of FACCipheringKey[KeyIndex] is less than 128, then FTCCipheringKey shall be the concatenation of zeros at the end (LSB) of FACCipheringKey[KeyIndex], such that the length of the result is 128.
 - The protocol shall perform the following:
 - Call the KeyStrengthRedAlg procedure specified in 3GPP2 S.S0078 with its inputs set as follows:
 - ♦ Set the *KeyLength* to 16.
 - ♦ Set the *OriginalKey* to the value of the FTCCipheringKey.
 - ♦ Set the *SaltLength* to the value of the FTCSaltLength parameter.
 - ♦ Set the *Salt* to the value of the FTCSalt parameter.
 - ♦ Set the *KeyEntropy* to the value of the FTCKeyEntropy parameter.
 - When the KeyStrengthRedAlg procedure returns, set the FTCCipheringKey to RedStrengthKey, which is the output of the KeyStrengthRedAlg procedure.
- If the value of the RTCCiphering attribute is equal to 0x01, then the protocol shall construct the Ciphering key for the Reverse Traffic Channel, RTCCipheringKey, as follows:
 - If the RACCipheringKey[KeyIndex] public data of the Key Exchange Protocol is set to NULL, the protocol shall set RTCCipheringKey to NULL.

- Otherwise, the protocol shall perform the following:
 - If the length of RACCipheringKey[KeyIndex] is equal to 128, then RTCCipheringKey shall be set to RACCipheringKey[KeyIndex].
 - Otherwise, if the length of RACCipheringKey[KeyIndex] is greater than 128, then RTCCipheringKey shall be the 128 most significant bits of RACCipheringKey[KeyIndex].
 - Otherwise, if the length of RACCipheringKey[KeyIndex] is less than 128, then RTCCipheringKey shall be the concatenation of zeros at the end (LSB) of RACCipheringKey[KeyIndex], such that the length of the result is 128.
- The protocol shall perform the following:
 - Call the KeyStrengthRedAlg procedure specified in 3GPP2 S.S0078 with its inputs set as follows:
 - ♦ Set the *KeyLength* to 16.
 - ♦ Set the *OriginalKey* to the value of the RTCCipheringKey.
 - ♦ Set the *SaltLength* to the value of the RTCSaltLength parameter.
 - ♦ Set the *Salt* to the value of the RTCSalt parameter.
 - ♦ Set the *KeyEntropy* to the value of the RTCKeyEntropy parameter
 - When the KeyStrengthRedAlg procedure returns, set the RTCCipheringKey to RedStrengthKey, which is the output of the KeyStrengthRedAlg procedure.

10.2.4.1.5 Constructing the Cryptosync

The protocol shall construct the Cryptosync as shown in Table 227.

Table 227—Subfield of the Cryptosync

Subfield	Length (bits)
FunctionCode	2
Reserved	6
Direction	1
RouteCounter	15
StreamID	16
SARResetCounter	8
VirtualSARSequenceNumber	48

FunctionCode	This field shall be set to '00' to indicate Ciphering function.
Reserved	The receiver shall ignore the values in this field.
Direction	If the payload being encrypted or decrypted is for Forward Link then this field shall be set to '0'. Otherwise, this field shall be set to '1'. Direction is received as Direction input to ENCRYPT and DECRYPT procedure calls.
RouteCounter	This field shall be set to the RouteCounter corresponding to the payload being encrypted or decrypted. RouteCounter is received as <i>RouteCounter</i> input to ENCRYPT and DECRYPT procedure calls.

StreamID	This field shall be set to the StreamID corresponding to the payload being encrypted or decrypted. StreamID is received as <i>StreamID</i> input to ENCRYPT and DECRYPT procedure calls.
SARResetCounter	This field shall be set to SARResetCounter corresponding to the payload being encrypted or decrypted. SARResetCounter is received as <i>SARResetCounter</i> input to ENCRYPT and DECRYPT procedure calls.
VirtualSARSequenceNumber	This field shall be set to VirtualSARSequenceNumber corresponding to the payload being encrypted or decrypted. VirtualSARSequenceNumber value is set as specified in subclauses 10.2.4.1.6 and 10.2.4.1.7.

10.2.4.1.6 Encrypt Procedures

The protocol shall provide encryption services through ENCRYPT procedure call.

If any of the following conditions is true:

- The Ciphering attribute for the channel under consideration (e.g., FTCCiphering) is equal to 0x00.
- The KeyIndex public data of the Key Exchange Protocol is set to NULL.
- The CipheringKey for the channel under consideration (e.g., FTCCipheringKey), constructed as specified in 10.2.4.1.4 where KeyIndex is set to KeyIndex public data of the Key Exchange Protocol, is NULL.

Then, the ENCRYPT procedure shall set the outputs of the ENCRYPT procedure as follows:

- Set *Payload* to the input *Payload*.
- Set *CipheringKeyIndex* to '00'.

Otherwise, the ENCRYPT procedure shall perform the following:

- If *DataUnit* is set to '01' then the ENCRYPT procedure shall perform the following:
 - The ENCRYPT procedure shall call the ESP_AES procedure specified in 3GPP2 S.S0055Av2.0 with its inputs set as follows:
 - Set the *key* to the CipheringKey for the channel under consideration (e.g., FTCCipheringKey) constructed as specified in 10.2.4.1.4 where KeyIndex is set to KeyIndex public data of the Key Exchange Protocol.
 - Set *fresh* to the value of the Cryptosync constructed as specified in 10.2.4.1.5, where VirtualSARSequenceNumber is set to (*SARSequenceRolloverCounter* | *SARSequenceNumber*).
 - Set the *freshsize* to the length of the Cryptosync in octets.
 - Set the *buf* to the address of the beginning of the memory space that contains the *Payload*.
 - Set the *bit_offset* to zero.
 - Set the *bit_count* to 8 times *PayloadSize*.
 - After the ESP_AES procedure is returned, the ENCRYPT procedure shall set the outputs of the ENCRYPT procedure as follows:
 - Set *Payload* to the output of the ESP_AES procedure that starts at the memory space specified by *buf* and is of the same size as the input *Payload*.
 - Set *CipheringKeyIndex* to KeyIndex public data of the Key Exchange Protocol.
- If *DataUnit* is set to '00' then the ENCRYPT procedure shall perform the following:

- Set `VirtualSARSequenceNumberInUse` to $(\text{SARSequenceRolloverCounter} \mid \text{SARSequenceNumber})$
- Set `n` to 1.
- The ENCRYPT procedure shall perform following steps *PayloadSize* times:³⁰
 - Call the ESP_AES procedure specified in 3GPP2 S.S0055Av2.0 with its inputs set as follows:
 - ♦ Set the key to the `CipheringKey` for the channel under consideration (e.g., `FTCCipheringKey`) constructed as specified in 10.2.4.1.4 where `KeyIndex` is set to `KeyIndex` public data of the Key Exchange Protocol.
 - ♦ Set `fresh` to the value of the `Cryptosync` constructed as specified in 10.2.4.1.5, where `VirtualSARSequenceNumber` is set to $16 \times \lfloor \text{VirtualSARSequenceNumberInUse} / 16 \rfloor$.
 - ♦ Set the *freshsize* to the length of the `Cryptosync` in octets.
 - ♦ Create a *temp_buf* and initialize it to $(\text{VirtualSARSequenceNumberInUse} \bmod 16)$ octets of 0x00 followed by the *nth* octet of *Payload*. Set the *buf* to the address of the beginning of the memory space that contains the *temp_buf*.
 - ♦ Set the *bit_offset* to zero.
 - ♦ Set the *bit_count* to $[(\text{VirtualSARSequenceNumberInUse} \bmod 16) + 1] \times 8$.
 - After the ESP_AES procedure is returned, the ENCRYPT procedure shall skip first $(\text{VirtualSARSequenceNumberInUse} \bmod 16)$ octets of the output of the ESP_AES procedure that starts at the memory space specified by *temp_buf*, and overwrite *nth* octet of the *Payload* with the value of the next octet of the ESP_AES procedure output.
 - Increment value of `n`
 - Increment value of `VirtualSARSequenceNumberInUse` modulo 2^{48} .
- The ENCRYPT procedure shall set the outputs of the ENCRYPT procedure as follows:
 - Set *CipheringKeyIndex* to `KeyIndex` public data of the Key Exchange Protocol.

10.2.4.1.7 Decryption procedures

The protocol shall provide decryption services through DECRYPT procedure call.

If the `Ciphering` attribute for the channel under consideration (e.g., `FTCCiphering`) is equal to 0x01 and *CipheringKeyIndex* is not set to '00', the DECRYPT procedure shall perform the following:

- If *DataUnit* is set to '01' then the DECRYPT procedure shall perform the following:
 - The DECRYPT procedure shall call the ESP_AES procedure specified in 3GPP2 S.S0055Av2.0 with its inputs set as follows:
 - Set the *key* to the `CipheringKey` for the channel under consideration (e.g., `FTCCipheringKey`) constructed as specified in 10.2.4.1.4 where `KeyIndex` is set to *CipheringKeyIndex*.
 - Set *fresh* to the value of the `Cryptosync` constructed as specified in 10.2.4.1.5, where `VirtualSARSequenceNumber` is set to $(\text{SARSequenceRolloverCounter} \mid \text{SARSequenceNumber})$.
 - Set the *freshsize* to the length of the `Cryptosync` in octets.

³⁰Even though steps here are performed on each octet of payload, in implementation the steps can be performed on data blocks of 16 octets of payload except first and last data blocks, which could have fewer octets.

- Set the *buf* to the address of the beginning of the memory space that contains the *Payload*.
- Set the *bit_offset* to zero.
- Set the *bit_count* to 8 times *PayloadSize*.
- After the ESP_AES procedure is returned, the DECRYPT procedure shall set the outputs of the DECRYPT procedure as follows:
 - Set *Payload* to the output of the ESP_AES procedure, which starts at the memory space specified by *buf* and is of the same size as the input *Payload*.
- If *DataUnit* is set to '00' then the DECRYPT procedure shall perform the following:
 - Set VirtualSARSequenceNumberInUse to (SARSequenceRolloverCounter | SARSequenceNumber).
 - Set *n* to 1.
 - The DECRYPT procedure shall perform following steps *PayloadSize* times:³¹
 - ♦ Call the ESP_AES procedure specified in 3GPP2 S.S0055Av2.0 with its inputs set as follows:
 - ♦ Set the key to the CipheringKey for the channel under consideration (e.g., FTCCipheringKey) constructed as specified in 10.2.4.1.4 where KeyIndex is set to CipheringKeyIndex.
 - ♦ Set *fresh* to the value of the Cryptosync constructed as specified in 10.2.4.1.5, where VirtualSARSequenceNumber is set to $16 \times \lfloor \text{VirtualSARSequenceNumberInUse} / 16 \rfloor$.
 - ♦ Set the *freshsize* to the length of the Cryptosync in octets.
 - ♦ Create a *temp_buf* and initialize it to (VirtualSARSequenceNumberInUse mod 16) octets of 0x00 followed by the *n*th octet of *Payload*. Set the *buf* to the address of the beginning of the memory space that contains the *temp_buf*.
 - ♦ Set the *bit_offset* to zero.
 - ♦ Set the *bit_count* to $[(\text{VirtualSARSequenceNumberInUse} \bmod 16) + 1] \times 8$.
 - After the ESP_AES procedure is returned, the DECRYPT procedure shall skip first (VirtualSARSequenceNumberInUse mod 16) octets of the output of the ESP_AES procedure that starts at the memory space specified by *temp_buf*, and overwrite *n*th octet of the *Payload* with the value of the next octet of the ESP_AES procedure output
 - Increment value of *n*.
 - Increment value of VirtualSARSequenceNumberInUse modulo 2^{48} .

Otherwise, the DECRYPT procedure shall set the outputs of the DECRYPT procedure as follows:

- Set *Payload* to the input *Payload*.

10.2.4.2 Message formats

No messages are defined for the InUse instance of this protocol.

10.2.4.3 Interface to other protocols

10.2.4.3.1 Commands

This protocol does not issue any commands.

³¹Even though steps here are performed on each octet of payload, in implementation the steps can be performed on data blocks of 16 octets of payload except first and last data blocks, which could have fewer octets.

10.2.4.3.2 Indications

This protocol does not register to receive any indications.

10.2.5 Configuration attributes

The following attributes and default values are defined (see 15.2.4 for the attribute record definition).

10.2.5.1 Simple attributes

The configurable simple attributes for this protocol are listed in the Table 228.

The default value for each attribute is typed in *bold italics*.

Table 228—Configurable values

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	FTCCiphering	Soft/Dynamic	0x00	RLP packets destined for the FTC shall not be encrypted by the sender and shall not be decrypted by the receiver.
			0x01	RLP packets destined for the FTC shall be encrypted by the sender and shall be decrypted by the receiver.
			0x02–0xff	Reserved.
0x0001	RTCCiphering	Soft/Dynamic	0x00	RLP packets destined for the RTC shall not be encrypted by the sender and shall not be decrypted by the receiver.
			0x01	RLP packets destined for the RTC shall be encrypted by the sender and shall be decrypted by the receiver.
			0x02–0xff	Reserved.

10.2.5.2 Complex attributes

The following complex attributes are defined for reduction of the Ciphering key strength for each channel.

10.2.5.2.1 FTCSaltLength attribute

The sender shall set AttributeID field to 0x8000 (see Table 229).

Table 229—FTCSaltLength attribute

Field	Length (bits)	Default value
FTCSaltLength	8	0
FTCSalt	FTCSaltLength × 8	N/A
FTCKeyEntropy	8	16

FTCSaltLength	The sender shall set this field to the length of the FTCSalt field in octets.
FTCSalt	The sender shall set this field to the value of the <i>Salt</i> input parameter that is to be used in the KeyStrengthRedAlg procedure specified in 3GPP2 S.S0078 for the FTC Ciphering key.
FTCKeyEntropy	The sender shall set this field to the value of the <i>KeyEntropy</i> input parameter that is to be used in the KeyStrengthRedAlg procedure specified in S.S0078 for the FTC Ciphering key. The valid values for this field are 0 through 16, inclusive.

Commit	Hard

Scope	Dynamic

10.2.5.2.2 RTCReducedStrengthCipheringKey attribute

The sender shall set AttributeID field to 0x8001 (see Table 230).

Table 230—RTCReducedStrengthCipheringKey attribute

Field	Length (bits)	Default value
RTCSaltLength	8	0
RTCSalt	RTCSaltLength × 8	N/A
RTCKeyEntropy	8	16

RTCSaltLength	The sender shall set this field to the length of the RTCSalt field in octets.
RTCSalt	The sender shall set this field to the value of the <i>Salt</i> input parameter that is to be used in the KeyStrengthRedAlg procedure specified in 3GPP2 S. for the RTC Ciphering key.
RTCKeyEntropy	The sender shall set this field to the value of the <i>KeyEntropy</i> input parameter that is to be used in the KeyStrengthRedAlg procedure specified in 3GPP2 S.S0078 for the RTC Ciphering key. The valid values for this field are 0 through 16, inclusive.

Commit	Hard

Scope	Dynamic

10.2.6 Non-attribute data

This protocol does not define any non-attribute data.

10.2.7 Protocol numeric constants

The protocol numeric constants are list in Table 231.

Table 231—Protocol numeric constants

Constant	Meaning	Value
N _{CPT} ype	Type field for this protocol	See 16.1
N _{CP} AES	Subtype field for this protocol	0x00

10.2.8 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

All configuration attributes and non-attribute data are Session State Information records. This protocol does not define additional parameter records.

10.3 Basic Message Integrity Protocol

10.3.1 Overview

The Basic Message Integrity Protocol provides a method for integrity protection of signaling messages by applying the AES CMAC function (see NIST, Special Publication 800-38B and IETF RFC 4493).

10.3.2 Primitives and public data

10.3.2.1 Commands

This protocol does not define any commands.

10.3.2.2 Return indications

This protocol does not define any indications.

10.3.2.3 Procedure calls

- AUTHENTICATE_ADD_TAG
 - Inputs: Direction, StreamID, RouteCounter, SARResetCounter, SARSequenceNumber, SARSequenceRolloverCounter, InputPayloadSize, InputPayload
 - Outputs: OutputPayloadSize, OutputPayload

Possible values of each input and output are as follows:

- *Direction* – '0' (ForwardLink) or '1' (ReverseLink)
- *StreamID* – 16-bit hexadecimal number
- *RouteCounter* – 15-bit hexadecimal number
- *SARResetCounter* – 8-bit hexadecimal number
- *SARSequenceNumber* – Hexadecimal number of n bits, where n is less than or equal to 48.
- *SARSequenceRolloverCounter* – Hexadecimal number of (48 – length of *SARSequenceNumber*) bits.
- *InputPayloadSize* – Input payload size in octets
- *InputPayload* – Input payload
- *OutputPayloadSize* – Output payload size in octets
- *OutputPayload* – Output payload
- AUTHENTICATE_CHECK_TAG
 - Inputs: Direction, StreamID, RouteCounter, SARResetCounter, SARSequenceNumber, SARSequenceRolloverCounter, InputPayloadSize, InputPayload
 - Outputs: TagMatched, OutputPayloadSize, OutputPayload

Possible values of each input and output are as follows:

- *Direction* – '0' (ForwardLink) or '1' (ReverseLink)
- *StreamID* – 16-bit hexadecimal number
- *RouteCounter* – 8-bit hexadecimal number
- *SARResetCounter* – 8-bit hexadecimal number
- *SARSequenceNumber* – Hexadecimal number of n bits, where n is less than or equal to 48.
- *SARSequenceRolloverCounter* – Hexadecimal number of (48 – length of *SARSequenceNumber*) bits.
- *InputPayloadSize* – Input payload size in octets
- *InputPayload* – Input payload
- *TagMatched* – '1' (TRUE), '0' (FALSE)
- *OutputPayloadSize* – Output payload size in octets
- *OutputPayload* – Output payload.
- CREATE_ID_TAG
 - Inputs: SequenceNumber
 - Outputs: IDTagSize, IDTag

Possible values of each input and output are as follows:

- SequenceNumber – 32-bit hexadecimal number
- IDTagSize – ID Tag size in octets
- IDTag – ID Tag

10.3.2.4 Local common data

This protocol does not define any local common data.

10.3.2.5 Public data

This protocol shall make the following data public:

- Subtype for this protocol
- All data defined as Static Attribute, Static Non-Attribute Data, and local common data

10.3.3 Protocol data unit

This protocol does not transmit or receive data. This protocol provides integrity protection services to Radio Link Protocol.

10.3.4 Procedures and messages for the InConfiguration instance of the protocol

10.3.4.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

10.3.4.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values (see 12.2).

10.3.4.3 Message formats

This protocol does not define any messages.

10.3.5 Procedures and messages for the InUse instance of the protocol

10.3.5.1 Procedures

10.3.5.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in 15.1.2.

10.3.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

10.3.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

10.3.5.1.4 Constructing the Message Integrity key

Basic Message Integrity Protocol shall construct the Message Integrity keys as follows:

- The protocol shall construct the Message Integrity key for the Forward Traffic Channel, FTCMIKey as follows:
 - If the FACMIKey[KeyIndex] public data of the Key Exchange Protocol is set to NULL, the protocol shall set FTCMIKey to NULL.
 - Otherwise, the protocol shall perform the following:
 - If the length of FACMIKey[KeyIndex] is equal to 128 bits, then FTCMIKey shall be set to FACMIKey[KeyIndex].
 - Otherwise, if the length of FACMIKey[KeyIndex] is greater than 128 bits, then FTCMIKey shall be the 128 most significant bits of FACMIKey[KeyIndex].
 - Otherwise, if the length of FACMIKey[KeyIndex] is less than 128 bits, then FTCMIKey shall be the concatenation of zeros at the end (LSB) of FACMIKey[KeyIndex], such that the length of the result is 128 bits.
- The protocol shall construct the Message Integrity key for the Reverse Traffic Channel, RTCMIKey as follows:
 - If the RACMIKey[KeyIndex] public data of the Key Exchange Protocol is set to NULL, the protocol shall set RTCMIKey to NULL.
 - Otherwise, the protocol shall perform the following:
 - If the length of RACMIKey[KeyIndex] is equal to 128 bits, then RTCMIKey shall be set to RACMIKey[KeyIndex].
 - Otherwise, if the length of RACMIKey[KeyIndex] is greater than 128 bits, then RTCMIKey shall be the 128 most significant bits of RACMIKey[KeyIndex].
 - Otherwise, if the length of RACMIKey[KeyIndex] is less than 128 bits, then RTCMIKey shall be the concatenation of zeros at the end (LSB) of RACMIKey[KeyIndex], such that the length of the result is 128 bits.

10.3.5.1.5 Constructing the Cryptosync

The protocol shall construct the Cryptosync as shown in Table 232.

Table 232—Subfield of the Cryptosync

Subfield	Length (bits)
Direction	1
RouteCounter	15
StreamID	16
SARResetCounter	8
VirtualSARSequenceNumber	48

Direction If the payload is for Forward Link then this field shall be set to '0'. Otherwise, this field shall be set to '1'. Direction is received as Direction input to AUTHENTICATE_ADD_TAG and AUTHENTICATE_CHECK_TAG procedure calls.

RouteCounter This field shall be set to the RouteCounter corresponding to the payload. RouteCounter is received as RouteCounter input to AUTHENTICATE_ADD_TAG and AUTHENTICATE_CHECK_TAG procedure calls.

StreamID This field shall be set to the StreamID corresponding to the payload. StreamID is received as StreamID input to AUTHENTICATE_ADD_TAG and AUTHENTICATE_CHECK_TAG procedure calls.

SARResetCounter This field shall be set to SARResetCounter corresponding to the payload. SARResetCounter is received as SARResetCounter input to AUTHENTICATE_ADD_TAG and AUTHENTICATE_CHECK_TAG procedure calls.

VirtualSARSequenceNumber This field shall be set to VirtualSARSequenceNumber corresponding to the payload. VirtualSARSequenceNumber value is set as specified in subclauses 10.3.5.1.7 and 10.3.5.1.8.

10.3.5.1.6 Authentication Header

The Authentication Header, which precedes the payload, has the format shown in Table 233.

Table 233—Authentication headers

Field	Length (bits)
AuthKeyIndex	2
Reserved	6
AuthTag	0 or 64

AuthKeyIndex The protocol shall set this field to indicate index of the key used for authentication of the payload as specified in Table 234.

Table 234—AuthKeyIndex encoding

AuthKeyIndex	Meaning
'00'	AuthTag not included
'01'	AuthTag computed with key corresponding to index '01'
'10'	AuthTag computed with key corresponding to index '10'
'11'	AuthTag computed with key corresponding to index '11'

Reserved The receiver shall ignore the values in this field.

AuthTag The protocol shall omit this field if the AuthKeyIndex field is set to '00'; otherwise, the protocol shall include this field and set it to authentication tag.

10.3.5.1.7 AUTHENTICATE_ADD_TAG procedures

The protocol shall provide service of adding authentication tag through AUTHENTICATE_ADD_TAG procedure call Figure 144 illustrates the relationship between an InputPayload and an OutputPayload of AUTHENTICATE_ADD_TAG procedure call.

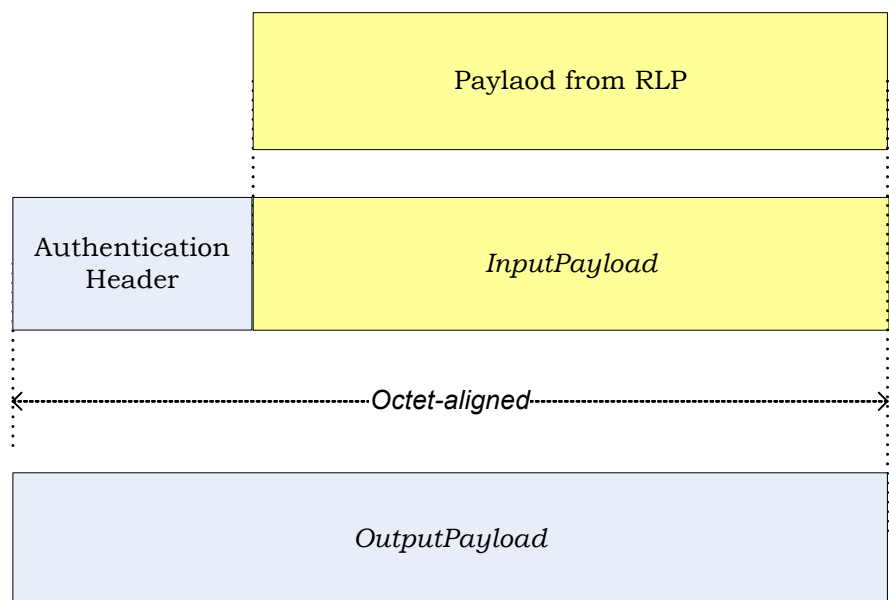


Figure 144—AUTHENTICATE_ADD_TAG procedure call payloads

If any of the following conditions is true:

- The KeyIndex public data of the Key Exchange Protocol is set to NULL.

- The MIKey for the channel under consideration (e.g., FTCTMIKey), constructed as specified in 10.3.5.1.4 where KeyIndex is set to KeyIndex public data of the Key Exchange Protocol, is NULL.
- Payload contains a message for which authentication tag is not required (See Clause 16 and each protocol text) and if protocol decides not to include the authentication tag.

then, the AUTHENTICATE_ADD_TAG procedure shall set the outputs of the AUTHENTICATE_ADD_TAG procedure as follows:

- Construct Authentication Header with the AuthKeyIndex field set to '00'.
- Set *OutputPayload* to (Authentication Header | *InputPayload*).
- Set *OutputPayloadSize* to (*InputPayloadSize* + size of the Authentication Header in octets).

Otherwise, the AUTHENTICATE_ADD_TAG procedure shall perform the following:

- The AUTHENTICATE_ADD_TAG procedure shall call the CMAC(K, M, Mlen, Tlen) procedure specified in 6.2 of NIST, Special Publication 800-38B with its inputs set as follows:
 - Set the K to the MIKey for the channel under consideration (e.g., FTCTMIKey) constructed as specified in 10.3.5.1.4 where KeyIndex is set to KeyIndex public data of the Key Exchange Protocol.
 - Set M to B0Header|B0| *InputPayload*, where B0Header and B0 are set as follows:
 - Set B0Header to '10011100'.³²
 - Set B0 to (fresh | L), where fresh is set to the value of the Cryptosync constructed as specified in 10.3.5.1.5 with VirtualSARSequenceNumber set (*SARSequenceRolloverCounter* | *SARSequenceNumber*), and L is set to size of the *InputPayload* in octets expressed as 32-bit hexadecimal number.
 - Set the Mlen to 8 times the size of M in octets.
 - Set the Tlen to 64.
- After the CMAC procedure is returned, the AUTHENTICATE_ADD_TAG procedure shall set the outputs of the AUTHENTICATE_ADD_TAG procedure as follows:
 - Construct Authentication Header with fields set as follows:
 - Set the AuthTag field to the output of the CMAC procedure.
 - Set the AuthKeyIndex field to KeyIndex public data of the Key Exchange Protocol.
 - Set *OutputPayload* to (Authentication Header | *InputPayload*).
 - Set *OutputPayloadSize* to (*InputPayloadSize* + size of the Authentication Header in octets).

10.3.5.1.8 AUTHENTICATE_CHECK_TAG procedures

The protocol shall provide service of checking authentication tag through AUTHENTICATE_CHECK_TAG procedure call Figure 145 illustrates the relationship between an InputPayload and an OutputPayload of AUTHENTICATE_CHECK_TAG procedure call.

³²Bits 5,4,3 of B0Header are encoded as (M-2)/2; where MSB of B0Header is bit 7, and M is Tlen/8 and can take the values 4, 6, 8, 10, 12, 14 and 16. Presently Tlen is always 64, and hence bits 5, 4, 3 of B0Header are always set to 011.

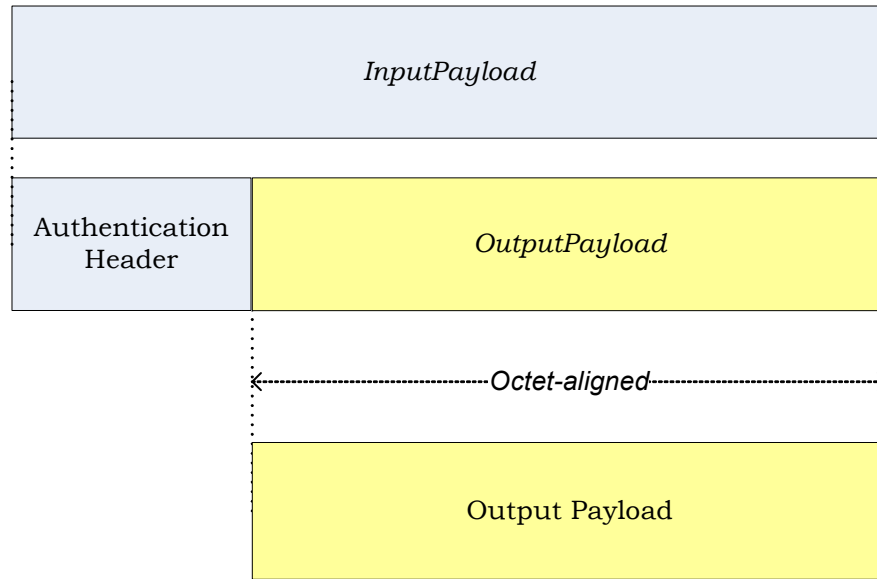


Figure 145—AUTHENTICATE_CHECK_TAG procedure call payloads

The AUTHENTICATE_CHECK_TAG procedure shall perform the following:

- The InputPayload consists of (Authentication Header | *OutputPayload*).
- The AUTHENTICATE_CHECK_TAG procedure shall remove Authentication Headers from *InputPayload* to produce *OutputPayload*.
- If the AuthKeyIndex field of the Authentication Header is set to '00', then AUTHENTICATE_CHECK_TAG procedure shall set the outputs of the AUTHENTICATE_CHECK_TAG procedure as follows:
 - Set *TagMatched* to TRUE.
 - Set *OutputPayloadSize* to size of *OutputPayload* in octets.
- Otherwise, the AUTHENTICATE_CHECK_TAG procedure shall call the CMAC(K, M, Mlen, Tlen) procedure specified in 6.2 of NIST, Special Publication 800-38B with its inputs set as follows:
 - Set the K to the MIKey for the channel under consideration (e.g., FTCMIKey) constructed as specified in 10.3.5.1.4 where KeyIndex is set to AuthKeyIndex field of the Authentication Header.
 - Set M to B0Header|B0| *OutputPayload*, where B0Header and B0 are set as follows:
 - Set B0Header to '10011100'.³³
 - Set B0 to (fresh | L), where fresh is set to the value of the Cryptosync constructed as specified in 10.3.5.1.5 with VirtualSARSequenceNumber set to (SARSequenceRolloverCounter | SARSequenceNumber), and L is set to size of the *OutputPayload* in octets expressed as 32-bit hexadecimal number.
 - Set the Mlen to 8 times size of M in octets.
 - Set the Tlen to 64.
- After the CMAC procedure is returned, the AUTHENTICATE_CHECK_TAG procedure shall set the outputs of the AUTHENTICATE_CHECK_TAG procedure as follows:

³³ Bits 5,4,3 of B0Header are encoded as (M-2)/2; where MSB of B0Header is bit 7, and M is Tlen/8 and can take the values 4, 6, 8, 10, 12, 14, and 16. Presently Tlen is always 64, and hence bits 5, 4, 3 of B0Header are always set to 011.

- If the output of the CMAC procedure matches the AuthTag field of the Authentication Header, then set *TagMatched* to TRUE. Otherwise, set *TagMatched* to FALSE.
- Set *OutputPayloadSize* to size of *OutputPayload* in octets.

10.3.5.1.9 CREATE_ID_TAG procedures

The protocol shall provide service of creating IDTag CREATE_ID_TAG procedure call.

The CREATE_ID_TAG procedure shall perform the following:

- If KeyIndex or RACMIKey[KeyIndex] public data of the Key Exchange Protocol is set to NULL, then set the outputs of the CREATE_ID_TAG procedure as follows:
 - Set *IDTagSize* to 0
 - Set *IDTag* to NULL
- Otherwise, perform the following:
 - Set MIKeyID to EHMAL-SHA256(key=RACMIKey[KeyIndex], message= "MIKeyID", MAC_length = 4), where RACMIKey[KeyIndex] and KeyIndex are public data of the Key Exchange Protocol and EHMAL-SHA256 function is specified in NIST, Special Publication 800-38B.
 - Set Tag to EHMAL-SHA256(key=RACMIKey[KeyIndex], message= *SequenceNumber*, MAC_length = 8), where RACMIKey[KeyIndex] and KeyIndex are public data of the Key Exchange Protocol, EHMAL-SHA256 function is specified in NIST, Special Publication 800-38B.
 - Set the outputs of the CREATE_ID_TAG procedure as follows:
 - Set *IDTagSize* to 12
 - Set *IDTag* to (MIKeyID | Tag)

10.3.5.2 Message formats

No messages are defined for the InUse instance of this protocol.

10.3.6 Interface to other protocols

10.3.6.1 Commands

This protocol does not issue any commands.

10.3.6.2 Indications

This protocol does not register to receive any indications.

10.3.7 Configuration attributes

This protocol does not define any configurable attributes.

10.3.8 Non-attribute data

This protocol does not define any non-attribute data.

10.3.9 Protocol numeric constants

Protocol numeric constants are shown in Table 235.

Table 235—Protocol numeric constants

Constant	Meaning	Value
N _{MIPT}	Type field for this protocol	See 16.1
N _{BMP}	Subtype field for this protocol	0x0000

10.3.10 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

All configuration attributes and Non-attribute data are Session State Information records. This protocol does not define additional parameter records.

10.4 Basic Key Exchange Protocol

10.4.1 Overview

The Basic Key Exchange Protocol provides a method for generating security keys at the access terminal and the access network based on a Pairwise Master Key. The Pairwise Master Key is established by higher layer protocols, such as EAP.

The Basic Key Exchange Protocol performs the following functions:

- Proves that both access terminal and access network have the same Pairwise Master Key.
- Derives fresh security keys from the Pairwise Master Key.
- Protects against a man-in-the-middle attack where a rogue entity causes the access terminal and the access network to agree upon a weaker security protocol.

10.4.2 Primitives and public data

10.4.2.1 Commands

This protocol does not define any commands.

10.4.2.2 Return indications

This protocol does not return any indications.

10.4.2.3 Local common data

This protocol defines the following local common data (i.e., AT only):

- PMKList[]
- A list of all Pairwise Master Keys.

10.4.2.4 Public data

This protocol shall make the following data public:

- Subtype for this protocol
- LatestPMK (access terminal only)
Latest PMK established or used on this route.
- LatestPMKCreationTime (access terminal only)
Creation time of the LatestPMK
- KeyIndex
The key index in use. Valid values are '01', '10' and '11'.
- FACMIKey[i] and its length for values of i '01' through '11'
The Message Integrity key for use on Forward Assigned Channels (e.g., the Forward Traffic Channel).
- RACMIKey[i] and its length for values of i '01' through '11'
The Message Integrity key for use on Reverse Assigned Channels (e.g., the Reverse Traffic Channel).
- FACCipheringKey[i] and its length for values of i '01' through '11'
The Ciphering key for use on Forward Assigned Channels (e.g., the Forward Traffic Channel).
- RACCipheringKey[i] and its length for values of i '01' through '11'
The Ciphering key for use on Reverse Assigned Channels (e.g., the Reverse Traffic Channel).
- All data defined as Static Attribute, Static Non-Attribute Data, and local common data

10.4.2.5 Interface to other protocols

10.4.2.5.1 Commands

This protocol does not define any commands.

10.4.2.5.2 Indications

This protocol does not register to receive any indications.

10.4.3 Protocol data unit

The transmission unit of this protocol is a message. This is a control protocol and, therefore, it does not carry payload on behalf of other layers or protocols.

This protocol uses the Signaling Protocol to transmit and receive messages.

10.4.4 Procedures and messages for the InConfiguration instance of the protocol

10.4.4.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

10.4.4.2 Procedures

This protocol uses the services of the Session Control Protocol to perform negotiation of attribute values

10.4.4.3 Message formats

This protocol does not define any messages.

10.4.5 Procedures and messages for the InUse instance of the protocol

10.4.5.1 Procedures

The Basic Key Exchange Protocol derives security keys and exchanges security capabilities as well as security protocols in use. The key exchange procedure uses the KeyRequest, KeyResponse, KeyComplete, KeyReject, and InitiateKeyRequest messages.

10.4.5.1.1 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and the access network shall perform the following in the order specified:

- Perform the procedures specified in 15.1.2.
- Access terminal shall Set LatestPMK to NULL.
- Access terminal shall set LatestPMKCreationTime to NULL.
- Set KeyIndex to NULL.
- Set FACMIKey[i] to NULL, for values of i '01' through '11'.
- Set RACMIKey[i] to NULL, for values of i '01' through '11'.
- Set FACCipheringKey[i] to NULL, for values of i '01' through '11'.
- Set RACCipheringKey[i] to NULL, for values of i '01' through '11'.

10.4.5.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

10.4.5.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4 when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

10.4.5.1.4 Access terminal requirements

When a Pairwise Master Key is established by higher layer protocols, the access terminal shall perform the following:

- Add the established PMK to the PMKList along with creation time of the key.
- Set LatestPMK to the established PMK
- Set LatestPMKCreationTime to creation time of the established PMK.

10.4.5.1.4.1 Initiating the key exchange

Access terminal may initiate the key exchange procedure by sending a KeyRequest message.

Upon receiving the InitiateKeyRequest message, the access terminal shall send a KeyRequest message unless access terminal has already sent a KeyRequest message and is awaiting the response.

10.4.5.1.4.2 Processing a KeyResponse message

Upon receiving the KeyResponse message with a TransactionID field that matches the TransactionID field of the associated KeyRequest message, the access terminal shall perform the following in the order in which the requirements are specified:

- The access terminal shall identify the PairwiseMasterKey from PMKList of any route that satisfies PairwiseMasterKeyID = EHMAC-SHA256(key = PairwiseMasterKey, message = “PairwiseMasterKeyID”, MAC_length = 8), where PairwiseMasterKeyID is a field of the received KeyResponse message, “PairwiseMasterKeyID” is the ASCII encoded value of the string, and the EHMAC-SHA256 function is specified in 10.4.5.1.8.
- If the access terminal cannot identify a valid PairwiseMasterKey that satisfies the above condition, then the access terminal shall send a KeyComplete message with Result set to 0x03, declare failure, and stop performing the rest of the key exchange procedure.
- The access terminal shall generate MICKey as specified in 10.4.5.1.6.
- The access terminal shall generate a MessageIntegrityCode as EHMAC-SHA256(key = MICKey, message = *Message*, MAC_length = 16), where *Message* is the received KeyResponse message with the MessageIntegrityCode field set to zero, and the EHMAC-SHA256 function is specified in 10.4.5.1.8.
- If the MessageIntegrityCode computed in the previous step does not match the MessageIntegrityCode field of KeyResponse message, then the access terminal shall declare failure, send a KeyComplete message with Result set to 0x01, and stop performing the rest of the key exchange procedure.
- If the supported ProtocolSetIdentifiers sent by the access network in the KeyResponse message do not match with the ProtocolSetIdentifiers supported by the access terminal (i.e., either all the ProtocolSetIdentifiers supported by the access terminal are not included in the KeyResponse message or KeyResponse message includes a ProtocolSetIdentifiers that is not supported by the access terminal), then the access terminal shall declare failure, send a KeyComplete message with Result set to 0x02, and stop performing the rest of the key exchange procedure.
- If the first ProtocolSetIdentifier field sent by the access network in the KeyResponse message do not match with the ProtocolSetIdentifier currently in use by the access terminal, then the access terminal shall declare failure, send a KeyComplete message with Result set to 0x04, and stop performing the rest of the key exchange procedure.
- The access terminal shall generate Message Integrity Keys and Ciphering Keys as specified in 10.4.5.1.7 for KeyIndex value specified in the KeyResponse message. The access terminal shall set KeyIndex public data to KeyIndex value specified in the KeyResponse message.
- The access terminal shall send a KeyComplete message with Result set to 0x00.
- If LatestPMKCreationTime is set to NULL or if creation time of PMK identified by the PairwiseMasterKeyID field of the received KeyResponse message is more recent than time indicated by LatestPMKCreationTime, then access terminal shall perform the following:
 - Set LatestPMK to PMK identified by the PairwiseMasterKeyID field of the received KeyResponse message.
 - Set LatestPMKCreationTime to creation time of LatestPMK.

10.4.5.1.4.3 Processing a KeyReject message

Upon receiving the KeyReject message with a TransactionID field that matches the TransactionID field of the associated KeyRequest message, the access terminal shall terminate the key exchange procedure.

10.4.5.1.5 Access network requirements

10.4.5.1.5.1 Initiating the key exchange

Access network may initiate the key exchange procedure by sending an InitiateKeyRequest message.

10.4.5.1.5.2 Processing a KeyRequest message

Upon receiving the KeyRequest message, the access network shall perform the following in the order in which the requirements are specified:

- The access network shall send a KeyResponse message or a KeyReject message.
- If a KeyResponse message was sent, the access network shall generate MICKey as specified in 10.4.5.1.6.

10.4.5.1.5.3 Processing a KeyComplete message

After receiving a KeyComplete message with a TransactionID field that matches the TransactionID field of the associated KeyResponse message, the access network shall perform the following:

- The access network shall generate a MessageIntegrityCode as EHMACHA256(key=MICKey, message = Message, MAC_length = 16), where Message is the received KeyComplete message with the MessageIntegrityCode field set to zero, and the EHMACHA256 function is specified in 10.4.5.1.8.
- If the MessageIntegrityCode computed in the previous step does not match the MessageIntegrityCode field of KeyComplete message, then the access network shall declare failure, and shall not use Message Integrity Keys and Ciphering Keys generated in this key exchange procedures.
- If the Result field of the KeyComplete message is not 0x00, then the access network shall declare failure and shall not use Message Integrity Keys and Ciphering Keys generated by this procedure.

10.4.5.1.6 MICKey derivation

The access terminal and the access network shall derive MICKey as follows:

- Set MICKey to zero and its length to 128.
- Set MICKey to EHMACHA256 (key = PairwiseMasterKey, message = "MICKey"||ATNonce|ANNonce, MAC_length = 16), where "MICKey" is the ASCII encoded value of the string, and the EHMACHA256 function is specified in 10.4.5.1.8.

10.4.5.1.7 Message Integrity key and Ciphering key generation

The access terminal and the access network shall generate a TSKey as specified in 10.4.5.1.7.1. The access terminal and the access network shall generate Message Integrity and Ciphering keys from TSKey as specified in 10.4.5.1.7.2.

10.4.5.1.7.1 Temporary security key derivation

The access terminal and the access network shall derive TSKey as follows:

- Set TSKey to zero and its length to $N_{\text{BKEPTKeyLen}}$.
- Set j to an 8-bit number with value zero.
- While $j < N_{\text{BKEPTKeyLen}}/256$
 - Set TSKey to $N_{\text{BKEPTKeyLen}}$ least significant bits of {TSKey | EHMACHA256(key = PairwiseMasterKey, message= "TSK" || ATNonce|ANNonce|j, MAC_length = 32)}, where "TSK" is the ASCII encoded value of the string, j is represented as an 8-bit field, and the EHMACHA256 function is specified in 10.4.5.1.8.
 - Set j to $j + 1$.

10.4.5.1.7.2 Message Integrity Key and Ciphering keys generation from TSKey

The keys used for Message Integrity and Ciphering are generated from the temporary security key using the procedures specified in this subclause.

The access network and the access terminal shall compute and store Message Integrity Keys and Ciphering Keys for KeyIndex *i* as follows:

- The access network and the access terminal shall set FACMIKey[*i*] to TSKey[127:0].
- The access network and the access terminal shall set RACMIKey[*i*] to TSKey[127:0].
- The access network and the access terminal shall set FACCipheringKey[*i*] to TSKey[255:128].
- The access network and the access terminal shall set RACCipheringKey[*i*] to TSKey[255:128].

10.4.5.1.8 EHMACHA256(key, message, MAC_length)

The EHMACHA256 procedure shall call ehmacsha256 procedure as specified in 3GPP2 S.S0078 with following inputs:

- Set the *key_length* to the length of the key in bits,
- Set the *key* to the key,
- Set the *message* to the message, and
- Set the *message_length* to the length of the message in bits,
- Set the *message_offset* to 0,
- Set the *MAC_length* to 8 × *MAC_length*.

The output of the EHMACHA256 function shall be set to the output of the ehmacsha256 procedure.

10.4.5.2 Message formats

10.4.5.2.1 KeyRequest

The access terminal sends the KeyRequest message to initiate the session key exchange (see Table 236).

Table 236—KeyRequest message

Field	Length (bits)
MessageID	8
TransactionID	8
ATNonce	128

- MessageID

The access terminal shall set this field to 0x00.
- TransactionID

The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment this field (modulo 256) for each successive message.
- ATNonce

The access terminal shall set this field to a 128-bit random number.

Channels	RTC	RLP	Reliable

Addressing	Unicast

AuthTag	Required when key is available

10.4.5.2.2 KeyResponse

The access network may send the KeyResponse message in response to the KeyRequest message (see Table 237).

Table 237— KeyResponse message

Field	Length (bits)
MessageID	8
TransactionID	8
KeyIndex	2
Reserved	6
PairwiseMasterKeyID	64
ANNonce	128
NumProtocolSetIdentifiers	8
NumProtocolSetIdentifiers occurrences of the following field:	
ProtocolSetIdentifier	16
MessageIntegrityCode	128

MessageID	The access network shall set this field to 0x01.
TransactionID	The access network shall set this field to the value of the TransactionID field of the KeyRequest message to which the access network is responding.
KeyIndex	The access network shall set this field to key index for which this key exchange is being initiated. Valid values for this field are '01' through '11'. The access network shall not set this field to KeyIndex public data.
Reserved	The access terminal shall ignore this field.
PairwiseMasterKeyID	The access network shall set this field to EHMAC-SHA256 (key = PairwiseMasterKey, message = "PairwiseMasterKeyID", MAC_length = 8), where PairwiseMasterKey is a PairwiseMasterKey to be used, "PairwiseMasterKeyID" is the ASCII encoded value of the string, and the EHMAC-SHA256 function is specified in 10.4.5.1.8.
ANNonce	The access network shall set this field to a 128-bit random number.
NumProtocolSetIdentifiers	The access network shall set this field to the number of ProtocolSetIdentifiers

supported by the access terminal. ProtocolSetIdentifiers supported by the access terminal are known to the access network through session information.

ProtocolSetIdentifier	The access network shall set this field to the ProtocolSetIdentifier supported by the access terminal. The access network shall include the ProtocolSetIdentifier currently in use at the beginning. Protocol subtypes supported by the access terminal are known to the access network through session information.
MessageIntegrityCode	The access network shall set this field to EHMAC-SHA256(key=MICKey, message = Message, MAC_length = 16), where Message is set to all fields of this message with this field set to zero, and the EHMAC-SHA256 function is specified in 10.4.5.1.8.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required when key is available

10.4.5.2.3 KeyComplete

The access terminal sends the KeyComplete message in response to the KeyResponse message (see Table 238).

Table 238—KeyComplete message

Field	Length (bits)
MessageID	8
TransactionID	8
Result	8
MessageIntegrityCode	128 ^a

^aThis field may be absent (see field description).

MessageID	The access terminal shall set this field to 0x02.
TransactionID	The access terminal shall set this field to the value of the TransactionID field of the corresponding KeyRequest message.
Result	The access terminal shall set this field according to Table 239.

Table 239—Definition of Result field

Value	Meaning
0x00	Key exchange successful.
0x01	MessageIntegrityCode failed
0x02	MessageIntegrityCode successful, but ProtocolSetIdentifiers verification failed.
0x03	PairwiseMasterKey not found.

Table 239—Definition of Result field (continued)

Value	Meaning
0x04	MessageIntegrityCode successful, ProtocolSetIdentifiers verification successful, but verification of ProtocolSetIdentifier in use failed.
All other values	Reserved.

MessageIntegrityCode If Result is 0x00, then the access terminal shall set this field to EHMAC-SHA256(key=MICKey, message = Message, MAC_length = 16), where Message is set to all fields of this message with this field set to zero, and the EHMAC-SHA256 function is specified in 10.4.5.1.8. Otherwise, the access terminal shall omit this field.

Channels	RTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required when key is available

10.4.5.2.4 KeyReject

The access network may send the KeyReject message in response to the KeyRequest message (see Table 240).

Table 240—KeyReject message

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access network shall set this field to 0x03.

TransactionID The access network shall set this field to the value of the TransactionID field of the KeyRequest message to which the access network is responding.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required when key is available

10.4.5.2.5 InitiateKeyRequest

The access network may send the InitiateKeyRequest message to request the access terminal to send a KeyRequest message (see Table 241).

Table 241—InitiateKeyRequest message

Field	Length (bits)
MessageID	8

MessageID The access network shall set this field to 0x04.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

10.4.5.3 Interface to other protocols

10.4.5.3.1 Commands

This protocol does not issue any commands.

10.4.5.3.2 Indications

This protocol does not register to receive any indications.

10.4.6 Configuration attributes

This protocol does not define any attributes.

10.4.7 Non-attribute data

This protocol does not define any non-attribute data.

10.4.8 Protocol numeric constants

Protocol numeric constants are listed in Table 242.

Table 242—Protocol numeric constants

Constant	Meaning	Value
N _{KEPT} Type	Type field for this protocol	See 16.1
N _{BKEP}	Subtype field for this protocol	0x0000
N _{BKEPTS} KeyLen	Length of Temporary Security Key in units of bits	256

10.4.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

All configuration attributes and non-attribute data are Session State Information records. This protocol defines the following parameter record in addition to the configuration attributes for this protocol.

10.4.9.1 PMK parameter

The sender shall set DataID field to 0x0000.

The Format of the Parameter Record for the PMK Parameter is listed in Table 243.

Table 243—Format of the parameter record for the PMK parameter

Field	Length (bits)
ValidPMKEExists	1
Reserved	7
PMKCount	8

PMKCount occurrences of the following five fields:

PMKLength	8
PMK	$\text{PMKLength} \times 8$
PMKGenerationTime	24
PMKMinLifeTime	24
PMKMaxLifeTime	24

ValidPMKEExists If a valid PMK exists, then this field shall be set to '1'. Otherwise this field shall be set to '0'.

Reserved The receiver shall ignore the values of this field.

PMKCount This field shall be set to the number of occurrences of the PMK field in this parameter record.

PMKLength This field shall be set to the length of the PMK field in units of octets.

PMK This field shall be set to a PairwiseMasterKey.

PMKGenerationTime This field shall be set to $X \bmod 2^{24}$, where X is the time, in units of seconds, at which the PairwiseMasterKey was generated.

PMKMinLifeTime This field shall be set to minimum lifetime of the PairwiseMasterKey in units of seconds.

PMKMaxLifeTime This field shall be set to maximum lifetime of the PairwiseMasterKey in units of seconds.

11. Connection Control Sublayer

11.1 Introduction

11.1.1 General overview

The protocols in the Connection Control Plane are control protocols, and do not carry data on behalf of other protocols. The protocols in this plane use the Signaling Protocol to transmit and receive messages (with the exception of the Overhead Messages Protocol that sends some information blocks using the Lower MAC Layer).

The Connection Control Plane controls the state of the air-link by managing the states of individual Lower MAC Layer protocols, and by providing individual Lower MAC Layer protocols with operating parameters.

There may be more than one InUse instance of the Connection Control Plane at the access terminal, with one instance for each access network that the access terminal communicates with. This chapter describes the interaction between one InUse instance at the access terminal and a corresponding InUse instance at one access network. The access terminal procedures are applicable only to the InUse instance being referred to. For example,

- “The access terminal shall transition to InActive State” shall be interpreted to mean that “This InUse instance of the access terminal shall transition to the InActive State.”
- “The access terminal has an assigned MAC ID” shall be interpreted to mean that “This InUse instance of the access terminal has an assigned MAC ID.”

In case a reference to more than one InUse instance is required, the reference will be stated explicitly. For example

- “If ExampleVariable is set to zero in all InUse instances at the access terminal, the access terminal shall issue an *Example* command,” shall be interpreted to mean that the instance of the protocol currently being referred to shall check ExampleVariable across all InUse instances, and issue the *Example* command if the condition is satisfied.

The access terminal and the access network maintain a connection whose state dictates the form in which communications between these entities can take place. The connection between the access terminal and the access network can be in one of the following four states:

- *Initialization State*: In this state, the access terminal acquires an access network.
- *Idle State*: In this state, the access terminal does not have Lower MAC resources assigned by the access network, and the access terminal may perform procedures related to page reception or access. This state is used when none of the InUse instances at the access terminal are in Connected State.
- *Suspended State*: In this state, the access terminal does not have Lower MAC resources assigned by the access network, and does not perform procedures related to page reception or access. This state is used when at least one of the InUse instances at the access terminal is in Connected State.
- *Connected State*: In this state, the access terminal has Lower MAC resources assigned by the access network.

The Connection Control Plane provides the following connection-related functions:

- Manages initial acquisition of the network.
- Manages opening and closing of connections.
- Maintains an approximate access terminal location in either connection states.
- Manages the radio link between the access terminal and the access network when a connection is open.

- Performs supervision at the access terminal both when the connection is open and when it is closed.

The Connection Control Plane performs these functions through the following protocols:

- *Air Link Management Protocol*: This protocol maintains the overall connection state in the access terminal and the access network. The protocol activates one of the following three protocols as a function of its current state.
- *Initialization State Protocol*: This protocol performs the actions associated with acquiring an access network.
- *Idle State Protocol*: This protocol performs the actions associated with an access terminal that has acquired the network, but does not have an assigned Lower MAC resource. Mainly, these are keeping track of the access terminal's approximate location in support of efficient Paging (with assistance from the Active Set Management protocol), the procedures leading to the opening of a connection, and support of access terminal power conservation.
- *Connected State Protocol*: This protocol performs the actions associated with an access terminal that has an open connection. These actions primarily include managing the radio link between the access terminal and the access network.
- *Active Set Management Protocol*: This protocol maintains an active set to provide an efficient radio link between the access terminal and the access network. This protocol is also responsible for keeping track of the access terminal's location, and the activation and deactivation of various Lower MAC Layer Protocols.
- *Overhead Messages Protocol*: This protocol broadcasts and receives essential parameters over the Superframe Preamble MAC and the Forward Traffic Channel MAC. This protocol also performs supervision on the parameters.

11.2 Basic Air Link Management Protocol

11.2.1 Overview

The Basic Air Link Management Protocol provides the following functions:

- General state machine and state-transition rules to be followed by an access terminal and an access network for the Connection Control Plane.
- Activation and deactivation of Connection Control Plane protocols applicable to each protocol state.
- Responding to supervision failure indications from other Protocols.
- Mechanism through which the access network can redirect the access terminal to another network.

The actual behavior and message exchange in each state is mainly governed by protocols that are activated by the Basic Air Link Management Protocol. These protocols return indications that trigger the state transitions of this protocol.

This protocol can be in one of four states:

- *Initialization State*: The protocol activates the Initialization State Protocol to execute the procedures relevant to this state. The access network does not support this state.
- *Idle State*: The protocol activates the Idle State Protocol to execute the procedures relevant to this state.
- *Suspended State*: There is no protocol dedicated to managing this state.
- *Connected State*: The protocol activates the Connected State Protocol to execute the procedures relevant to this state.

Figure 146 provides an overview of the access terminal states and state transitions. All transitions are caused by indications returned from protocols activated by the Basic Air Link Management Protocol.

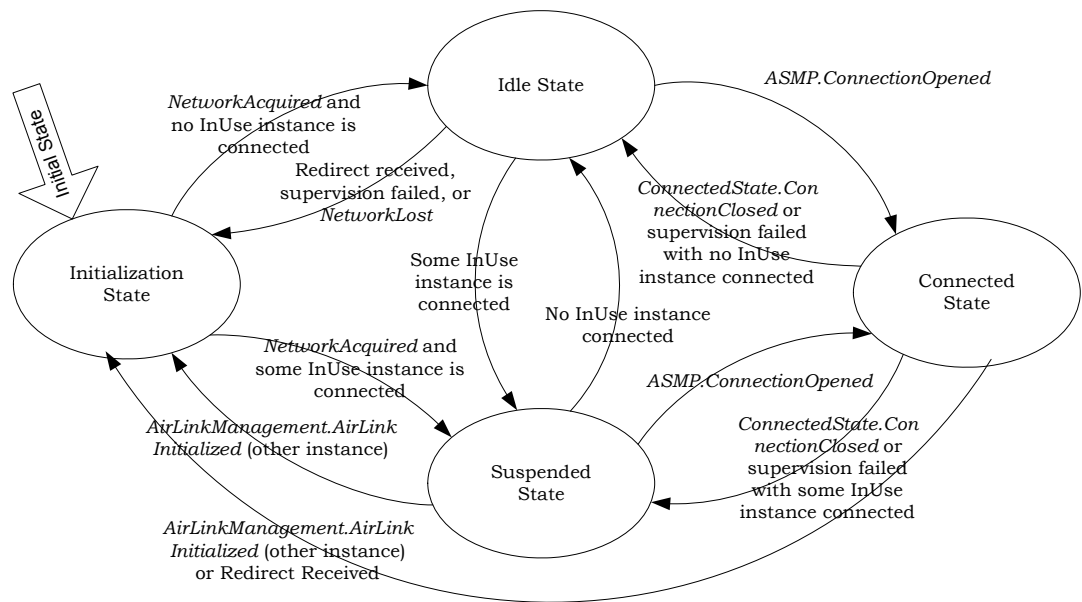


Figure 146—Air Link Management Protocol State Diagram (access terminal)

Figure 147 provides an overview of the access network states and state transitions. The access network transitions between the Suspended and Idle States are outside the scope of this standard, and may depend on backhaul messaging between multiple access networks.

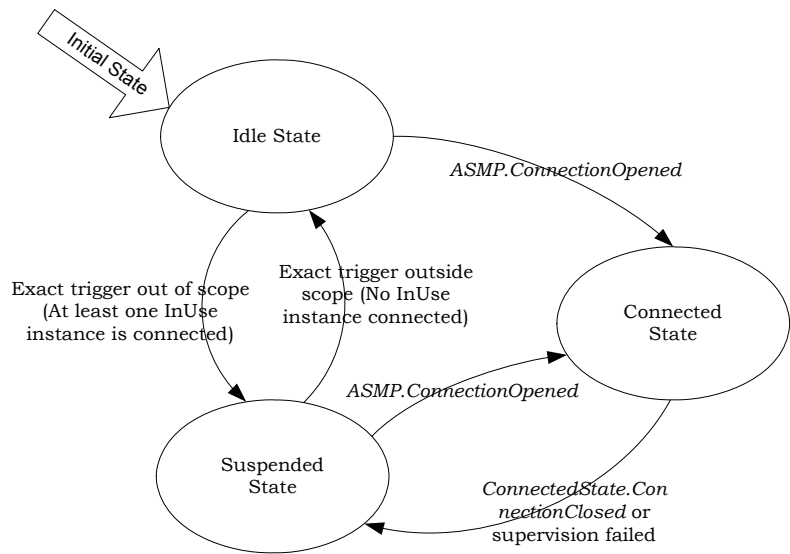


Figure 147—Air Link Management Protocol State Diagram (access network)

Table 244 provides a summary of the Connection Control Plane and Lower MAC Sublayer protocols that are active in each state.

Table 244—Active Protocols Per Air Link Management Protocol State

Protocol	Initialization State ^a	Suspended State	Idle State	Connected State
Overhead Messages Protocol	X	X	X	X
Initialization State Protocol	X		X	X
Active Set Management Protocol		X (Connected)	X (Idle)	X (Connected)
Superframe Preamble MAC Protocol	X	X ^a	X	X
Forward Link Control Segment MAC Protocol		X ^b	X	X
Forward Traffic Channel MAC Protocol		X ^b	X	X
Access Channel MAC Protocol		X ^b	X	X ^b
Reverse Traffic Channel MAC Protocol		X ^b		X
Reverse Control Channel MAC Protocol		X ^b		X

^aApplicable only to access terminal.^bUsed by access terminal only during handoff to an asynchronous sector and exit from semi-connected state.

11.2.2 Primitives

11.2.2.1 Commands

This protocol defines the following commands:

- *OpenConnection*
- *CloseConnection*
- *SendRegistration*

11.2.2.2 Return indications

This protocol returns the following indications:

- *ConnectionClosed*
- *AirLinkInitialized*
- *IdleInitiated*
- *TunedAway*
- *TunedBack*
- *ConnectionOpened*
- *RegistrationSucceeded*

11.2.2.3 Local common data

- *ConnectionFailureRecord*
- *TransactionID* for last sent *ConnectionFailureRecord*
- *CurrentRegisteredZone*
- *RegistrationFailed*
- *SmallRegistrationDistanceEnabled*

- LastRegisteredSector
 - PrimaryRegistrationZoneCode of LastRegisteredSector
 - Latitude and Longitude of LastRegisteredSector
 - NetworkCode of LastRegisteredSector

11.2.2.4 Public data

- Subtype for this protocol
- State of this protocol
- PowerDownRegistrationPerformed
- Frames where the access terminal performs tune-away
- TuneAwayStatus

11.2.3 Procedures and messages for the InConfiguration instance of the protocol

11.2.3.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

11.2.3.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

11.2.3.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

11.2.4 Procedures for the InUse instance

11.2.4.1 Initialization and commit procedures

11.2.4.1.1 Initialization procedures

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol shall set PowerDownRegistrationPerformed to 0.
- The protocol at the access terminal shall enter the Initialization State.
- The protocol at the access network shall enter the Idle State.

11.2.4.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

11.2.4.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

11.2.4.2 Command processing

11.2.4.2.1 OpenConnection

The procedures for processing this command are described under the procedures for the respective protocol states.

11.2.4.2.2 CloseConnection

The *CloseConnection* command causes the access terminal to relinquish dedicated Lower MAC resources the access terminal currently has, and to terminate ongoing steps to acquire dedicated Lower MAC resources.

If the protocol receives the *CloseConnection* command in the Idle State:

- Access terminal shall issue an *IdleState.Close* command.

If the protocol receives the *CloseConnection* command in the Connected State:

- Access terminal and access network shall issue a *ConnectedState.CloseConnection* command to one of the InUse instances at which the Air Link Management Protocol is in the Connected State.

If the protocol receives this command in any other state it shall be ignored.

11.2.4.2.3 SendRegistration

The access terminal shall ignore this command if this InUse instance is not the session anchor instance. Otherwise, the access terminal shall send a Registration Message and set public SmallRegistrationDistanceEnabled = 0.³⁴

11.2.4.3 Initialization State

This state is not applicable to the access network. In the Initialization State the access terminal has no information about the serving access network. In this state the access terminal selects a serving access network and obtains time synchronization from the access network.

11.2.4.3.1 Access terminal requirements

This protocol at the access terminal shall enter the Initialization State when the Basic Air Link Management Protocol is instantiated.

If the protocol receives the *OpenConnection* command, the access terminal shall queue the command.

This protocol may enter the Initialization State on events such as network redirection and initial power-on. A comprehensive list of events causing the Basic Air Link Management Protocol to enter the Initialization State is beyond the scope of this standard.

Upon entering the Initialization State, when the access terminal is using more than one InUse instance, and the Initialization procedure has been completed in one of the InUse instances, the access terminal may perform the following two steps and omit the remaining procedures required in this subclause.

- If the Air Link Management Protocol in at least one other InUse instance is in Connected State, the access terminal shall transition to the Suspended State

³⁴If the access network receives a Registration message, the access network should assume that the access terminal has SmallRegistrationDistanceEnabled = 0.

- Otherwise, the access terminal shall transition to the Idle State

Upon entering the Initialization State, the access terminal shall:

- Issue an *ActiveSetManagement.Deactivate* command.
- Issue an *OverheadMessages.Deactivate* command.
- Issue an *IdleState.Deactivate* command.
- Issue a *ForwardTrafficChannelMAC.Deactivate* command.
- Issue a *ForwardLinkControlSegmentMAC.Deactivate* command.
- Issue an *InitializationState.Activate* command. If the access terminal entered this state because the protocol received a Redirect message and a ChannelBand Record was received with the message, the access terminal shall provide the ChannelBand Record with the command.
- If the protocol receives an *InitializationState.NetworkAcquired* indication, the access terminal shall perform the following: Issue a *InitializationState.Deactivate* command.
- If the Air Link Management Protocol in at least one other InUse instance is in Connected State, the access terminal shall transition to the Suspended State.
- Otherwise, the access terminal shall transition to the Idle State.

11.2.4.3.2 Access network requirements

This state is not defined for the access network.

11.2.4.4 Idle state

In this state the access terminal has acquired the access network but does not have an open connection with any access network. In this state, the access terminal performs page reception, registration, and access procedures.

11.2.4.4.1 Access terminal requirements

The access terminal shall perform the following operations upon entering this state:

- Issue an *IdleState.Activate* command
- Issue an *ActiveSetManagement.Activate* command
- Generate an *AirLinkManagement.IdleInitiated* indication

If the protocol receives an *ActiveSetManagement.ConnectionOpened* indication, the access terminal shall

- Issue an *IdleState.Deactivate* command
- Transition to the Connected State

If the access terminal receives an *AirLinkManagement.ConnectionOpened* indication from any other InUse instance, the access terminal shall

- Issue an *IdleState.Deactivate* command
- Transition to the Suspended State

If the access terminal had a queued *OpenConnection* command, or receives this command in the Idle State, the access terminal shall process the command as follows. The access terminal shall first determine if the connection can be opened using this InUse instance. If the cached overhead messages indicate that the Pilot in the Active Set maintained by the Active Set Management Protocol is known to belong to this InUse instance,

- The access terminal shall issue an *IdleState.OpenConnection* command.

Otherwise,

- The access terminal shall generate a *RouteControl.CreateNewStack(PilotID)* command where PilotID is the Pilot in the Active Set.
- The access terminal shall send an *OpenConnection* command to the Air Link Management Protocol of the newly created InUse instance.

If the protocol receives a Redirect message, an *ActiveSetManagement.NetworkLost* indication or an *OverheadMessages.SupervisionFailed* indication, the access terminal shall

- Set the public data LastRegisteredSector to NULL.
- Return an *AirLinkInitialized* indication.
- Transition to the Initialization State.

If the access terminal receives an *AirLinkManagement.AirLinkInitialized* indication from another InUse instance, the access terminal shall transition to the Initialization State.

11.2.4.4.1.1 Procedures for registration

In Idle State, the access terminal shall perform registration procedures for zone-based, distance-based, timer-based, NetworkCode-based, or loss of coverage-based registration.

11.2.4.4.1.1.1 Zone-based registration

In the SectorParameters, each sector shall advertise a PrimaryRegistrationZoneCode, and zero or more SecondaryRegistrationZoneCodes. The access terminal shall store the PrimaryRegistrationZoneCode of the LastRegisteredSector.

The access terminal shall perform the following operation after receiving an *ActiveSetManagement.IdleHandoff* or *OverheadMessages.SectorParametersUpdated* indication. If the PrimaryRegistrationZoneCode of the LastRegisteredSector does not match either the PrimaryRegistrationZoneCode or any of the SecondaryRegistrationZoneCodes in the SectorParameters message from the current Active Set member, the access terminal shall generate an *AirLinkManagement.SendRegistration* command to the session anchor InUse instance.

11.2.4.4.1.1.2 Distance-based registration

The access terminal shall perform the following operation after receiving an *ActiveSetManagement.IdleHandoff* or *OverheadMessages.SectorParametersUpdated* indication:

- If (x_0, y_0) are the longitude and latitude of the sector LastRegisteredSector, and (x_n, y_n) are the longitude and latitude (in units of $\frac{1}{4}$ seconds) of the sector currently in the Active Set of the access terminal, then r is given by the following:³⁵

³⁵The x 's denote longitude and the y 's denote latitude. The access terminal shall compute r with an error of no more than $\pm 5\%$ of its true value when $|y_L/14400|$ is less than 60 and with an error of no more than $\pm 7\%$ of its true value when $|y_L/14400|$ is between 60 and 70.

$$r = \left\lceil \frac{\sqrt{\left[D_{\text{longitude}} \times \cos\left(\frac{\pi}{180} \times \frac{y_o}{14400}\right) \right]^2 + [y_n - y_o]^2}}{16} \right\rceil$$

$$D_{\text{longitude}} = x_n - x_o \text{ if } \left| \frac{x_n - x_o}{14400} \right| < 180$$

$$D_{\text{longitude}} = 360 \times 14400 - |x_n - x_o| \text{ if } \left| \frac{x_n - x_o}{14400} \right| \geq 180$$

- If SmallRegistrationDistanceEnabled is set to '0', the access terminal shall generate an *AirLinkManagement.SendRegistration* command to the session anchor InUse instance if the computed value r is greater than the value provided in the RegistrationRadiusLarge attribute.
- If SmallRegistrationDistanceEnabled is set to '1', the access terminal shall generate an *AirLinkManagement.SendRegistration* command to the session anchor InUse instance if the computed value of r is greater than the values of the RegistrationRadiusSmall attribute.

11.2.4.4.1.1.3 Timer-based registration

The access terminal performs timer-based registration in accordance with the procedures described by Session keep-alive function of the Session Control Plane.

11.2.4.4.1.1.4 NetworkCode-based registration

The access terminal shall perform the following operation after receiving an *ActiveSetManagement.IdleHandoff* or *OverheadMessages.SectorParametersUpdated* indication:

- If the NetworkCode field of the current sector is different from the NetworkCode of the LastRegisteredSector, the access network shall generate an *AirLinkManagement.SendRegistration* command to the session anchor InUse instance.

11.2.4.4.1.1.5 Registration after loss of coverage

If InitializationStateRegistrationTimeout is greater than zero, and the access terminal enters Idle State after remaining in Initialization State for more than InitializationStateRegistrationTimeout minutes, the access terminal shall generate an *AirLinkManagement.SendRegistration* command to the session anchor InUse instance.

This requirement does not prevent an access terminal from generating an *AirLinkManagement.SendRegistration* command if the access terminal was in Initialization State for less than InitializationStateRegistrationTimeout minutes.

11.2.4.4.1.2 Procedures for maintaining LastRegisteredSector

Upon entering Idle State, the access terminal shall set LastRegisteredSector as follows:

- If all the following conditions are satisfied, the access terminal shall set LastRegisteredSector to the sector that was FLSS the time Connected State was exited.
- RegistrationFailed is equal to '0'

- The Route Control Protocol is in Open State for the InUse instance corresponding to at least one pilot in the Combined Active Set. The Combined Active Set is public data of the Active Set Management Protocol.
- The Connected State Protocol at some InUse instances has transitioned to Open State during this connection.

Otherwise, the access terminal shall not change LastRegisteredSector.

11.2.4.4.2 Access network requirements

The access network shall issue the following commands upon entering this state:

- IdleState.Activate
- ActiveSetManagement.Activate

If the access network receives an *OpenConnection* command in the Idle State, the access network shall

- Issue an *IdleState.OpenConnection* command.

If the protocol receives an *ActiveSetManagement.ConnectionOpened* indication, the access network shall

- Issue an *IdleState.Deactivate* command.
- Transition to the Connected State.

The access network should transition to the Suspended State when the access terminal is in Connected State on any of the InUse instances. At the access network, the exact trigger for transition from the Idle State to Suspended State is outside the scope of this standard. When the access network transitions to the Suspended State, the access network shall issue an *IdleState.Deactivate* command.

- The access network may send the access terminal a Redirect message to redirect it from the current serving network and optionally, provide it with information directing it to another network.

11.2.4.5 Suspended State

In the Suspended State, the access terminal does not have a Lower MAC resource assigned by the access network.

11.2.4.5.1 Access terminal requirements

The access terminal shall issue the following command upon entering this state:

- ActiveSetManagement.Open

If the access terminal instance receives an *AirLinkManagement.IdleInitiated* indication the access terminal shall transition to the Idle State.

If the access terminal receives an *ActiveSetManagement.ConnectionOpened* indication, the access terminal shall transition to the Connected State.

If the access terminal receives an *AirLinkManagement.AirLinkInitialized* indication from another InUse instance, the access terminal shall transition to the Initialization State.

If this protocol receives a *Signaling.ReliableSignalingStreamReset* indication, it shall issue a *AirLinkManagement.CloseConnection* command.

The access terminal shall ignore the *OpenConnection* command in this state.

11.2.4.5.2 Access network requirements

The access network shall issue the following command upon entering this state:

- *ActiveSetManagement.Open*

If the access network receives an *ActiveSetManagement.ConnectionOpened* indication, the access network shall transition to the Connected State.

The access network should transition to the Idle State when the access terminal is not in Connected State at any of the InUse instances. At the access network, the exact trigger for transition from the Suspended State to the Idle State is outside the scope of this standard. For example, in case the session anchor access network is not the serving access network, then the session anchor access network may be informed through backhaul messaging about supervision failure at the serving access network. Such information may cause the session anchor access network to transition from the Suspended to Idle State and initiate paging procedures for the access terminal if necessary.

11.2.4.6 Connected State

In the Connected State, the access terminal has Lower MAC resources assigned by the access network.

11.2.4.6.1 Access terminal requirements

11.2.4.6.1.1 General requirements

The access terminal shall perform the following upon entering this state:

- Return a *ConnectionOpened* indication
- Issue a *ConnectedState.Activate* command
- Issue an *ActiveSetManagement.Open* command

If the access terminal receives an *OpenConnection* command in this state, it shall issue a *ConnectedState.ExitSemiConnectedState* command.

If this protocol receives a *Signaling.ReliableSignalingStreamReset* indication or a *Signaling.ReliableMessageDeliveryFailed* indication, it shall issue a *AirLinkManagement.CloseConnection* command.

If the protocol receives a *ConnectedState.ConnectionClosed*, an *ActiveSetManagement.EmptyMACResourceAssigned*, an *OverheadMessages.SupervisionFailed*, a *ForwardTrafficChannelMAC.SupervisionFailed*, *ReverseControlChannelMAC.SupervisionFailed* or a *ReverseTrafficChannelMAC.SupervisionFailed* indication, the access terminal shall

- Perform the cleanup procedure defined in 11.2.4.6.1.3.
- If the Air Link Management Protocol at one or more of the InUse instances is in Connected State, the access terminal shall transition to the Suspended State. Otherwise, the access terminal shall issue a *ReverseTrafficChannelMAC.Deactivate* and a *ReverseControlChannelMAC.Deactivate* command, and transition to the Idle State.

If the protocol receives a Redirect message, or an *AirLinkManagement.AirLinkInitialized* indication from another InUse instance, the access terminal shall

- Perform the cleanup procedure defined in 11.2.4.6.1.3.
- Issue a *ReverseTrafficChannelMAC.Deactivate* and a *ReverseControlChannelMAC.Deactivate* command.

- Return an *AirLinkInitialized* indication.
- Transition to the Initialization State.

11.2.4.6.1.2 Connection failure processing

If the access terminal receives an *OverheadMessages.SupervisionFailed* indication and if the value of the *ConnectionFailureReportingEnabled* attribute is 0x01, then the access terminal shall generate and store a *ConnectionFailure* record that contains the reason for connection failure, the System Time when the Connection failure occurred, the SectorID public data of the Overhead Messages Protocol at the time the Connection failure occurred, the ChannelBand on which the Connection failure occurred, and, if available, the Latitude, and Longitude of the location where the connection failure occurred. The access terminal shall store at least four Connection failure records corresponding to the last four connection failures.

11.2.4.6.1.3 Connected State cleanup procedures

The access terminal shall return a *ConnectionClosed* indication and issue the following commands when it exits this state:

- *ConnectedState.Deactivate*

11.2.4.6.2 Access network requirements

11.2.4.6.2.1 General requirements

The access network shall issue the following commands upon entering this state:

- *ConnectedState.Activate*
 - If the transition to Connected State is due to the reception of an access probe, this command shall be accompanied by *ConnectType = Access* flag.
 - If the transition to Connected State is due to the sending of a *MACResourceAssignment*, this command shall be accompanied by *ConnectType = ActiveSetAdd* flag.
- *ActiveSetManagement.Open*

If the access network receives an *OpenConnection* command in this state, it shall issue a *ConnectedState.ExitSemiConnectedState* command.

If the protocol receives a *ConnectedState.ConnectionClosed*, or *ActiveSetManagement.ConnectionLost* indication, the access network shall

- Perform the cleanup procedures defined in 11.2.4.6.2.2.
- Transition to the Suspended State.

The access network may send the access terminal a Redirect message to redirect it from the current serving network and optionally, provide it with information directing it to another network. If the access network sends a Redirect message it shall

- Perform the cleanup procedures defined in 11.2.4.6.2.2.
- Transition to the Suspended State.

11.2.4.6.2.1.1 Connection Failure Processing

The access network shall send a *ConnectionFailureReportAck* message upon receiving a *ConnectionFailureReport* message.

11.2.4.6.2.2 Connected State cleanup procedures

The access network shall issue the following commands when it exits this state:

- ConnectedState.Deactivate
- ReverseTrafficChannelMAC.Deactivate
- ReverseControlChannelMAC.Deactivate

11.2.4.7 Common procedures

11.2.4.7.1 KeepAlive procedures

In Connected State and Suspended State, the access terminal shall send a ConnectionKeepAlive message every ConnectionKeepAlivePeriod seconds.

The access network may use this message to supervise resources it has assigned to the access terminal.

11.2.4.7.2 Procedures for maintaining registration

If the access terminal sends a Registration message, it shall start a registration response timer with duration RegistrationTimeout.

The access terminal shall check the validity of RegistrationResponse messages as follows. If the TransactionID in a RegistrationResponse message is different from the last Registration message sent by the access terminal, the RegistrationResponse message shall be discarded. Otherwise, the RegistrationResponse message shall be processed.

While the registration response timer is active

- If the access terminal gets a RegistrationResponse message with RegistrationResponseType = RouteOpenRequired, the access terminal shall
 - Disable the registration response timer
 - Set public data RegistratonFailed to '1'
 - Send a Registration message
- If the registration response timer expires, the access terminal shall
 - Set public data RegistratonFailed to '1'
 - Send a Registration message
- If the access terminal gets a RegistrationResponse message with RegistrationResponseType = Success, or gets a *RouteControl.UATIAssigned* indication, the access terminal shall
 - Disable the registration response timer
 - Set LastRegisteredSector to the FLSS
 - Set public data RegistrationFailed to '0'
 - Generate a *RegistrationSucceeded* indication

If the access terminal receives a RegistrationModeUpdate message, the access terminal shall set SmallRegistrationDistanceEnabled equal to the field included in the received message.

11.2.4.7.3 Procedures for Power Down registration

11.2.4.7.3.1 Access terminal procedures

The access terminal shall perform the following procedures in Connected State and Suspended State.

On transition to the Connected State or Suspended State

- If the transition was from the Idle State and if PowerDownRegistrationPerformed is equal to 1 in some InUse instance, the access terminal should cancel the power down registration by sending a PowerDownRegistration message with PowerDownStatus set to 0.
- Otherwise, if PowerDownRegistrationPerformed is equal to 1 in some InUse instance, the access terminal shall send a PowerDownRegistration message with the PowerDownStatus set to 1, and shall set PowerDownRegistrationPerformed to 1.

If the access terminal sends a PowerDownRegistration message with PowerDownStatus field equal to 1, the access terminal shall

- Set PowerDownRegistrationPerformed to 1.
- Send a PowerDownRegistration message with PowerDownStatus set to 0 on all existing InUse instances.

If the access terminal sends a PowerDownRegistration message with the PowerDownStatus field equal to 0, the access terminal shall

- Set PowerDownRegistrationPerformed to 0.
- Send a PowerDownRegistration message with PowerDownStatus field set to 0 on all existing InUse instances.

11.2.4.7.3.2 Access network procedures

If the access network receives a PowerDownRegistration message, the access network should set PowerDownRegistrationPerformed to the value of the PowerDownStatus field in the message.

11.2.4.7.4 Connection failure reporting

These procedures for reporting connection failures are common to Suspended and Connected States.

If the access terminal has stored ConnectionFailure records and if the value of the ConnectionFailureReportingEnabled attribute is 0x01, and if this InUse instance is the session anchor, then the access terminal shall send a ConnectionFailureReport message upon entering Connected or Suspended state. Upon receiving the corresponding ConnectionFailureReportAck message, the access terminal shall delete the reported ConnectionFailure records. For each ConnectionFailureReport message that is sent, the access terminal shall increment the TransactionID that is part of the local common data.

If the access terminal does not receive a ConnectionFailureReportAck message in response to a ConnectionFailureReport message, then the access terminal may retransmit the ConnectionFailureReport message an implementation specific number of times.

11.2.4.7.5 TuneAway procedures

TuneAway defines a repetitive set of time periods during which the access terminal and access network do not exchange any transmission. TuneAway is used by the access terminal and access network in Idle or Connected States to allow the access terminal to receive transmission on other ChannelBands or neighbor technologies.

The beginning and end of TuneAway is determined by the TuneAway attribute and SectorOffset that is calculated from the SubSuperframeOffset and SuperframeOffset fields of PilotReport messages sent by the access terminal in Connected State. Further, the access terminal and access network may operate on multiple tune away schedules. Each tune away schedule is specified by a separate TuneAway attribute.

The tune away operation is controlled through a variable TuneAwayStatus, that is public data of the protocol.

If the TuneAwayStatus is set to '1' at the access terminal:

- The access terminal may stop monitoring the forward channels and shall stop transmitting on the reverse channels.

If the TuneAwayStatus is set to '1' at the access network:

- The access network may stop monitoring the reverse channels and shall stop transmitting to this access terminal on the forward channels.

11.2.4.7.5.1 TuneAway time calculations

The following formulas determine the beginning and end of tune away periods for a tune away schedule as a function of the TuneAway attribute.

Consider tune away schedule N and consider a given serving sector. Let t_0 be the system time at the beginning of StartSuperframeNumberN. Then, TuneAway period number n for a sector with given SectorOffset begins at

$$\text{TuneAwayTime}_{N_n} = t_0 + \text{StartSuperframeOffset}_N + \text{SectorOffset} + (n-1) \times \text{TuneAwayPeriod}_N$$

and ends at

$$\text{TuneBackTime}_{N_n} = \text{TuneAwayTime}_{N_n} + \text{TuneAwayDuration}_N$$

Here SectorOffset is set to the SectorOffset corresponding to the serving sector in the last PilotReport message sent by the access terminal. SectorOffset shall be set to zero if the access terminal has not sent any SectorOffset field, or if the access terminal is in Idle State. The SectorOffset field may be used by the access terminal to shift tune away schedules for asynchronous sectors, depending on the offset between the asynchronous sector and the system the access terminal is tuning away to.

The SectorOffset field is computed from the SubSuperframeOffset and SuperframeOffset fields of the PilotReport as follows, where $T_{\text{SUPERFRAME}}$ is the superframe duration defined in the Physical Layer.

$$\text{SectorOffset} = \text{SubSuperframeOffset} + (2^{15} - \text{SuperframeOffset}) \times T_{\text{SUPERFRAME}}$$

Since a TuneAwayTime and a TuneBackTime may be misaligned with PHY frame boundaries, actual tune away and tune back operations obey the following parameters.

$\text{TuneAwayFrame}_{N_n}$ shall be set to the PHY frame number of the reverse link PHY frame that contains time instance $\text{TuneAwayTime}_{N_n}$. In case $\text{TuneAwayTime}_{N_n}$ lies on a boundary of PHY frames j and j+1, $\text{TuneAwayFrame}_{N_n}$ shall be set to j.

$\text{TuneBackFrame}_{N_n}$ shall be set to the PHY frame number of the reverse link PHY frame that contains time instance $\text{TuneBackTime}_{N_n}$. In case $\text{TuneBackTime}_{N_n}$ lies on a PHY frame boundary of PHY frames j and j + 1, $\text{TuneBackFrame}_{N_n}$ shall be set to j + 1.

For each tune away schedule, the access terminal and access network may be in one of two states:

- *Camped State*: When a tune away schedule is in this state, it does not require the access terminal to tune away.
- *TunedAway State*: When a tune away schedule is in this state, it allows the access terminal to tune away.

11.2.4.7.6 Procedures for setting TuneAwayStatus

The access network and access terminal shall enter the Camped state for a tune away schedule N if any of the following conditions are met:

- At the beginning of PHY frame number $\text{TuneBackFrame}N_n$

The access network and access terminal shall enter the TunedAway state for a tune away schedule N

- At the end of PHY frame number $\text{TuneAwayFrame}N_n - 1$

The access terminal shall set the public data TuneAwayStatus as follows:

- If any of the tune away schedules is in TuneAway state, this protocol shall set TuneAwayStatus to '1'.
- Otherwise, the access terminal shall set TuneAwayStatus to '0'.
- If TuneAwayStatus changes from '1' to '0', this protocol shall generate a *TunedBack* indication.

The access network shall set the public data TuneAwayStatus as follows:

- If any of the tune away schedules is in TuneAway state, this protocol shall set TuneAwayStatus to '1'.
- Otherwise, the access network shall set TuneAwayStatus to '0'.
- If TuneAwayStatus changes from '1' to '0', this protocol shall generate a TunedBack indication.

In Idle State, the access terminal and access network shall perform this calculation only for those TuneAwaySchedules for which IdleStateApplicable is equal to 1.

If TuneAwayStatus changes from '0' to '1', this protocol shall generate a *TunedAway* indication.

11.2.5 Message formats for the InUse instance

11.2.5.1 Redirect

The access network sends the Redirect message to redirect the access terminal(s) away from the current network; and, optionally, the access network provides it with information directing it to one of a set of different networks (see Table 245).

Table 245—Redirect message

Field	Length (bits)
MessageID	8
StayAwayDuration	16
RedirectReason	4
NumChannelBands	4

Table 245—Redirect message (continued)

Field	Length (bits)
NumChannelBands instances of the following field:	
ChannelBand	Variable

MessageID	The access network shall set this field to 0x00.
StayAwayDuration	The access network shall set this field to the duration, in units of seconds, for which the access terminal should not make an access attempt at the sector sending this message. The value 0xff means that the access terminal may select an implementation dependent time.
RedirectReason	The sender shall set this field to reflect the redirect reason, as shown in Table 246.

Table 246—Encoding of the RedirectReason field

Field value	Description
0x0	Reserved
0x1	Network Busy
0x2	Authentication or billing failure
0x3	Desired QoS unavailable
0x4	No backhaul connectivity
0x5	Network Maintenance
0xf	General Failure
All other values are reserved	

NumChannelBands	The access network shall set this field to the number of ChannelBand records it is including in this message.
ChannelBand	This field shall be set to the ChannelBand record of the channel that the access terminal should reacquire. The channel band shall be specified using the standard ChannelBand record definition (see 15.2.1).

Channels	FTC	
Addressing	Broadcast	Unicast

RLP	Best Effort	
AuthTag	Not required	

11.2.5.2 Registration

The access terminal sends the Registration message to the access network to support paging and tracking for Idle State (see Table 247).

Table 247—Registration message

Field	Length (bits)
MessageID	8
TransactionID	8
RegistrationReason	4
Reserved	4

MessageID The access terminal shall set this field to 0x01.

TransactionID The sender shall set this field to zero if this is the first instance of this message sent by this InUse instance, and shall increment the field (modulo 256) for each successive message.

RegistrationReason The access terminal shall set this field to indicate a reason for the registration (see Table 248).

Table 248—RegistrationReason encoding

PowerDownRegistrationReason field value	Meaning
0x0	Reason unspecified
0x1	Zone-based
0x2	Distance based
0x3	NetworkID based
0x4	Registration after loss of coverage
0x5	Session keep alive initiated by the Session Layer (timer-based registration)
All other values	Reserved

Reserved The access network shall ignore this field.

Channels	RTC
Addressing	Unicast

RLP	Best Effort
AuthTag	Required if key is available

11.2.5.3 RegistrationResponse

The access network sends the RegistrationResponse message to the access terminal in response to the Registration message (see Table 249).

Table 249—RegistrationResponse message

Field	Length (bits)
MessageID	8
TransactionID	8
RegistrationResponseType	4
Reserved	4

MessageID The access terminal shall set this field to 0x02.

TransactionID The access network shall set this field to the same value as the TransactionID in the received Registration message.

RegistrationResponseType The access network shall set this field to indicate the type of response to the registration message sent by the access terminal (see Table 250).

Table 250—RegistrationResponseType value

RegistrationResponseType	Meaning
0x0	Registration successful
0x1	RouteOpen required
All other values	Reserved

Reserved The access terminal shall ignore this field.

Channels	FTC

RLP	Best Effort

Addressing	Unicast

AuthTag	Required if key is available

11.2.5.4 RegistrationModeUpdate

The access network sends the RegistrationModeUpdate message to control distance based registration procedures at the access terminal (see Table 251).

Table 251—RegistrationModeUpdate message

Field	Length (bits)
MessageID	8
SmallRegistrationDistanceEnabled	1
Reserved	7

MessageID The access network shall set this field to 0x03.

SmallRegistrationDistanceEnabled

The access network shall set this field to '1' to cause the access terminal to register when the registration distance between the current sector and the last registered sector is greater than RegistrationRadiusSmall.

Reserved The access terminal shall ignore this field.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key is available

11.2.5.5 ConnectionFailureReport

The ConnectionFailureReport message format is as follows in Table 252.

Table 252—ConnectionFailureReport message

Field	Length (bits)
MessageID	8
TransactionID	8
NumRecords	4
NumRecords occurrences of the following record:	
ConnectionFailureReason	4
TimeStamp	37
SectorID	128
ChannelBand	Variable
LatLongIncluded	1
Latitude	0 or 22
Longitude	0 or 23
Padding	0 – 7 (as needed)

MessageID The access terminal shall set this field to 0x04.

TransactionID The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment this field (modulo 256) for each successive message.

NumRecords The access terminal shall set this field to the number of Connection failures reported in this message.

ConnectionFailureReason This field indicates the reason for the connection failure. The access terminal shall set this field according to Table 253.

Table 253—ConnectionFailureReason Encoding

ConnectionFailureReason field value	Meaning
0x0	General Connection Failure (excluding failure due to a cause to which a specific code-point has been assigned)
All other values	See 3GPP2 C.R1001-F v1.0

TimeStamp The access terminal shall set this field to the system time at which this Connection failure occurred in units of superframes.

SectorID The access terminal shall set this field to the SectorID field of the SectorParameters message transmitted by the access network where the Connection failure occurred. SectorID is public data of the Overhead Messages Protocol.

ChannelBand The access terminal shall set this field to the ChannelBand record (see 15.2.1) for the ChannelBand on which the Connection failure occurred.

LatLongIncluded The access terminal shall set this field to '1' if the Latitude and Longitude fields are included in this message. Otherwise, the access terminal shall set this field to '0'.

Latitude If LatLongIncluded is set to '0', then the access terminal shall omit this field. Otherwise, the access terminal shall set this field to the Latitude of the location where the Connection failure occurred.

Longitude If LatLongIncluded is set to '0', then the access terminal shall omit this field. Otherwise, the access terminal shall set this field to the Longitude of the location where the Connection failure occurred.

Padding The access terminal shall add Padding bits to make the length of the entire message an integer number of octets. The access terminal shall set these bits to '0'. The access network shall ignore this field.

Channels	RTC

RLP	Best Effort

Addressing	Unicast

AuthTag	Required if key is available

11.2.5.6 ConnectionFailureReportAck

The ConnectionFailureReportAck message format is as follows in Table 254.

Table 254—ConnectionFailureReportAck message

Field	Length (bits)
MessageID	8

Table 254—ConnectionFailureReportAck message (continued)

Field	Length (bits)
TransactionID	8

MessageID The sender shall set this field to 0x07.

TransactionID The sender shall set this value to the TransactionID field of the corresponding ConnectionFailureReport message.

Channels	FTC
Addressing	Broadcast Unicast

RLP	Best Effort
AuthTag	Required if key is available

11.2.5.7 ConnectionKeepAlive

The access terminal sends the ConnectionKeepAlive message periodically to indicate that the access terminal is in Connected or Suspended state (see Table 255).

Table 255—ConnectionKeepAlive message

Field	Length (bits)
MessageID	8
RLActivity	1
Reserved	7

MessageID The access terminal shall set this field to 0x05.

RLActivity The access terminal shall set this field to 1 if it has transmitted any reverse link data (excluding air interface signaling messages) on any Route since the previous keep alive message was sent.

Reserved The access network shall ignore this field.

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required if key is available

11.2.5.8 PowerDownRegistration

The access terminal sends the PowerDownRegistration message to help the access network control paging behavior (see Table 256).

Table 256—PowerDownRegistration message

Field	Length (bits)
MessageID	8
MessageSequence	8

Table 256—PowerDownRegistration message (continued)

Field	Length (bits)
PowerDownStatus	1
PowerDownRegistrationReason	2
Reserved	5

- MessageID** The access terminal shall set this field to 0x06.
- MessageSequence** The access terminal shall set this field to zero if this is the first Registration message sent by this InUse instance, and shall increment the field (modulo 2^S , $S = 8$) for each successive message.
- PowerDownStatus** The access terminal shall set this field to 1 to indicate that it will not perform page reception in Idle State. The access terminal shall set this field to 0 to indicate that it will perform page reception in Idle State.
- PowerDownRegistrationReason**
The access terminal shall set this field to indicate a reason for power down registration (see Table 257).

Table 257—PowerDownRegistrationReason encoding

PowerDownRegistrationReason field value	Meaning
00	This value shall be used if PowerDownStatus is equal to 0.
01	Battery Critically Low ^a
10	User initiated power down
All other values	Reserved

^aThe AT can send a PowerDownRegistration message with the PowerDownRegistrationReason set to '01' when the AT determines that the battery is critically low, but there is enough power left to continue the connection for an undetermined amount of time. By waiting for supervision failure to close the connection, the AN can maintain the connection until the AT battery can no longer continue the connection.

- Reserved** The access network shall ignore this field.

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required if key is available

11.2.6 Interface to other protocols

11.2.6.1 Commands

This protocol issues the following commands:

- *InitializationState.Activate* (AT only)

- *InitializationState.Deactivate* (AT only)
- *IdleState.Activate*
- *IdleState.Deactivate*
- *IdleState.Close*
- *IdleState.OpenConnection*
- *ConnectedState.Activate*
- *ConnectedState.Deactivate*
- *ConnectedState.CloseConnection*
- *ConnectedState.ExitSemiConnectedState*
- *ActiveSetManagement.Activate*
- *ActiveSetManagement.Deactivate* (AT only)
- *ActiveSetManagement.Open*
- *IdleState.OpenConnection*
- *OverheadMessages.Deactivate* (AT only)
- *ReverseTrafficChannelMAC.Deactivate*
- *ReverseControlChannelMAC.Deactivate*
- *ForwardLinkControlSegmentMAC.Deactivate* (AT only)
- *ForwardTrafficChannelMAC.Deactivate* (AT only)
- *RouteControl.CreateNewStack* (AT only)

11.2.6.2 Indications

This protocol registers to receive the following indications:

- *AirLinkManagement.AirLinkInitialized* (AT only, from other InUse instances)
- *AirLinkManagement.ConnectionOpened* (AT only, from other InUse instances)
- *InitializationState.NetworkAcquired* (AT only)
- *ActiveSetManagement.ConnectionOpened*
- *ConnectedState.ConnectionClosed*
- *ConnectedState.GlobalConnectionCloseRequired*
- *ActiveSetManagement.ConnectionLost* (AN only)
- *ActiveSetManagement.NetworkLost* (AT only)
- *OverheadMessages.SupervisionFailed* (AT only)
- *ReverseTrafficChannelMAC.SupervisionFailed* (AT only)
- *ReverseControlChannelMAC.SupervisionFailed* (AT only)
- *RouteControl.RouteOpened* (AN only)
- *ActiveSetManagement.IdleHandoff* (AT only)
- *OverheadMessages.SectorParametersUpdated* (AT only)
- *Signaling.ReliableSignalingStreamReset*

11.2.7 Configuration attributes

11.2.7.1 Simple attributes

The access network and the access terminal shall use the default values that are typed in ***bold italics*** (see Table 258).

Table 258—Configurable simple attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	ConnectionFailureReportingEnabled	Soft Allowed/Static	0x00	Access terminal does not send ConnectionFailureReport messages.
			0x01	Access terminal sends ConnectionFailureReport messages.
			All other values	Reserved
0x0001	InitializationStateRegistrationTimeout	Soft Allowed/Static	0x0a	Value of this field in minutes
			0x00	Access terminal does not perform registration based on time out in InitializationState
			All other values	The value of this field in minutes
0x0002	ConnectionKeepAlivePeriod	Soft Allowed/Dynamic	0x00	The access terminal does not send ConnectionKeepAlive messages.
			0x0a	Value of this attribute in seconds
			0x20–0xff	Reserved
			All other values	Value of this attribute in seconds
0x0003	RegistrationTimeout	Soft Allowed/Dynamic	0x03	Value of this attribute in seconds
			0x00 and 0x0b–0xff	Reserved
			All other values	Value of this attribute in seconds

11.2.7.2 Complex attributes

11.2.7.2.1 TuneAwayScheduleN attribute

This is a static attribute. N takes values from 0 through $N_{\text{ALMPTuneAwayMaxSched}} - 1$.

This complex attribute shall determine the periodicity, duration and offset of tuneaways that the access terminal may perform. Such tuneaways may be used for handoff candidate search or alternate technology page reception.

This attribute has AttributeID 0x810N (see Table 259).

Table 259—TuneAwayScheduleN attribute

Field	Length (bits)	Default
IdleStateApplicable	2	0
StartSuperframeNumber	37	0
StartSuperframeOffset	16	0
TuneAwayDuration	24	0
TuneAwayPeriod	32	0
Reserved	1	0

IdleStateApplicable This tune away schedule will be applicable in Idle State if this field is 1, and will not be applicable if this field is 0. Other values of this field are reserved.

StartSuperframeNumber To compute the tuneaway cycles, it shall be assumed that the first tuneaway occurred in this superframe.

StartSuperframeOffset This field is a measure of time in units of 1 micro second. To compute the tuneaway cycles, it shall be assumed that the first tuneaway begins StartSuperframeOffset time after the beginning of superframe number StartSuperframeNumber.

TuneAwayDuration This field determines the duration of the tune away in units of 1 micro second.

TuneAwayPeriod This field determines the time between the start of successive tuneaways in units of 1 microsecond.

Reserved This field shall be ignored by the receiver.

Commit	Soft allowed

Scope	Dynamic

11.2.7.2.2 RegistrationRadius

This complex attribute shall determine the registration distance used by the access terminal for distance based registration.

This attribute has AttributeID 0x8000 (see Table 260).

Table 260—RegistrationRadius attribute

Field	Length (bits)	Default
RegistrationRadiusSmall	16	0
RegistrationRadiusLarge	16	0x0800

RegistrationRadiusSmall This field shall be set to the registration radius the access terminal is required to use when SmallRegistrationDistanceEnabled=1.

RegistrationRadiusLarge This field shall be set to the registration radius the access terminal is required to use when SmallRegistrationModeEnabled=0.

Commit	Soft allowed	Scope	Dynamic

11.2.8 Protocol numeric constants

Protocol numeric constants are listed in Table 261.

Table 261—Protocol numeric constants

Constant	Meaning	Value
N _{ALMPT} Type	Type field for this protocol	See 16.1
N _{ALMP} Basic	Subtype field for this protocol	0x00
N _{ALMPT} TuneAwayMaxSched	Maximum number of tune away schedules	0x04

11.2.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

11.3 Basic Initialization State Protocol

11.3.1 Overview

The Basic Initialization State Protocol provides the procedures and messages required for an access terminal to acquire a serving network. This protocol imposes the following two requirements on an access terminal

- The access terminal follows channel band information provided in a redirect in the Air Link Management Protocol.
- Prevents the access terminal from attempting to connect to an access network with a protocol stack that is not supported by the access network (as defined through InitialProtocolSetIdentifiers).

At the access network, this protocol does not define any states.

At the access terminal, this protocol operates in one of the following four states:

- *Inactive State*: In this state the protocol waits for an *Activate* command.
- *Network Determination State*: In this state the access terminal chooses an access network on which to operate.
- *Pilot Acquisition State*: In this state the access terminal acquires a Forward Pilot Channel.
- *Read SystemInfo State*: In this state the access terminal reads the SystemInfo block and determines if the InitialProtocolSetIdentifier fields in the SystemInfo block can be supported by the access terminal.

Protocol states and events causing transition between states are shown in Figure 148, where the three steps of Initialization are defined.

When the access terminal is using more than one InUse instance, and the Initialization procedure has been completed in one of the InUse instances, the other InUse instances may proceed through the Initialization procedure instantaneously upon Protocol activation, based on buffered information.

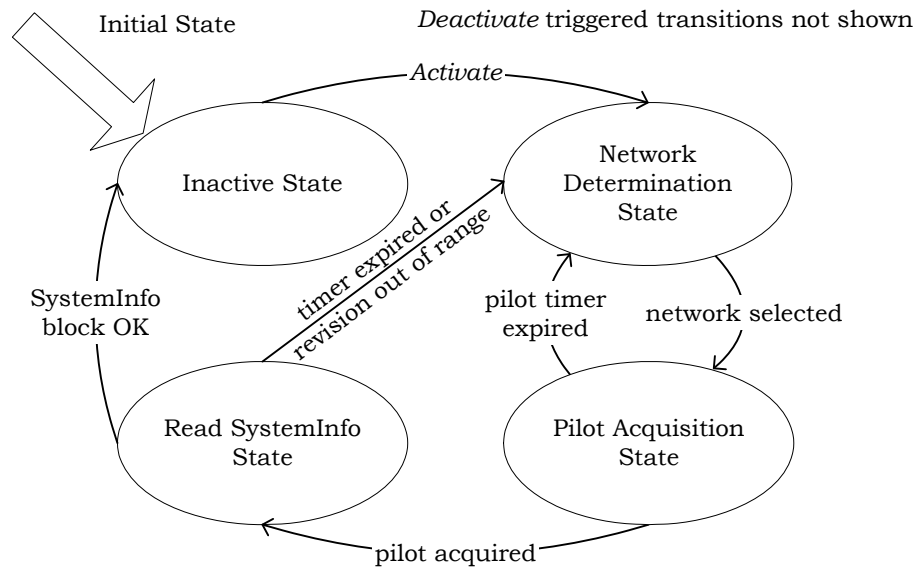


Figure 148—Basic Initialization State Protocol State Diagram (access terminal)

11.3.2 Primitives and public data

11.3.2.1 Commands

This protocol at the access terminal defines the following commands:

- *Activate* (an optional ChannelBand record can be specified with the command)
- *Deactivate*

11.3.2.2 Return indications

This protocol at the access terminal returns the following indications:

- *NetworkAcquired*

11.3.2.3 Local common data

This protocol does not define any local common data.

11.3.2.4 Public data

This protocol defines the following public data:

- Selected ChannelBand
- System time (in asynchronous systems, the system time may be different for different InUse instances)
- PilotID of the selected sector
- The InitialProtocolSetIdentifier (one or more values) of the SystemInfo block

11.3.3 Procedures and messages for the InConfiguration instance of the protocol

11.3.3.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

11.3.3.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

11.3.3.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

11.3.4 Procedures for the InUse instance

11.3.4.1 Initialization and commit procedures

11.3.4.1.1 Initialization procedures

Upon creation, the InUse instance of this protocol in the access terminal shall perform the following:

- The access terminal shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the Inactive State.

11.3.4.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

11.3.4.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

11.3.4.2 Command processing

The access network shall ignore all commands.

11.3.4.2.1 Activate

If the protocol receives an *Activate* command in the Inactive State, the access terminal shall transition to the Network Determination State.

If the protocol receives an *Activate* command in any other state, the access terminal shall ignore it.

11.3.4.2.2 Deactivate

If the protocol receives a *Deactivate* command in the Inactive State, the access terminal shall ignore it.

If the protocol receives a *Deactivate* command in any other state, the access terminal shall transition to the Inactive State.

11.3.4.3 Inactive State

In the Inactive State the access terminal waits for the protocol to receive an *Activate* command.

11.3.4.4 Network Determination State

In the Network Determination State, the access terminal selects a ChannelBand and attempts to acquire the access network.

If a ChannelBand record was provided with the *Activate* command, the access terminal should select the system and channel specified by the record. Such a record may be provided by the access network when the Initialization State Protocol is activated due to a redirect received by the Air Link Management Protocol.

The specific mechanisms to provision the access terminal with a list of preferred networks and with the actual algorithm used for network selection are beyond the scope of this standard.

Upon selecting a ChannelBand the access terminal shall enter the Pilot Acquisition State.

11.3.4.5 Pilot Acquisition State

In the Pilot Acquisition State, the access terminal acquires the pilot of the selected ChannelBand using the SuperframePreamble MAC.

Upon entering the Pilot Acquisition State, the access terminal shall tune to the selected ChannelBand and shall search for the pilot. If the access terminal acquires the pilot, it shall enter the Read SystemInfo State. If the access terminal fails to acquire the pilot within $T_{ISPPilotAcq}$ seconds of entering the Pilot Acquisition State, it shall enter the Network Determination State.

11.3.4.6 Read SystemInfo State

Upon entering this state, the access terminal shall issue the *SuperframePreambleMAC.Activate* and *OverheadMessages.Activate* commands.

If the access terminal fails to receive *OverheadMessages.SystemInfoReceived* indication, within $T_{ISPSyncAcq}$ seconds of entering the Read SystemInfo State, the access terminal shall issue *SuperframePreambleMAC.Deactivate* and *OverheadMessages.Deactivate* commands and shall enter the Network Determination State. While attempting to receive the SystemInfo block, the access terminal shall discard any other messages received on the Control Channel.

When the access terminal receives a SystemInfo block:

- If the access terminal is unable to support any of the InitialProtocolSetIdentifier fields specified in the block, the access terminal shall issue *SuperframePreambleMAC.Deactivate* and *OverheadMessages.Deactivate* commands and enter the Network Determination State.
- Otherwise, the access terminal shall
 - Set the access terminal time to the time specified in the SystemInfo block. The time specified in the block is the time at the beginning of the superframe where the block transmission started. This time is accurate only to the nine least significant digits, and is enough for the access terminal to read the ExtendedChannelInfo message (on the Forward Traffic Channel MAC) that contains the complete time.
 - Return a *NetworkAcquired* indication.
 - Enter the Inactive State.

11.3.5 Message formats for the InUse instance

The InUse instance of this protocol does not define any messages.

11.3.6 Interface to other protocols

11.3.6.1 Commands

This protocol at the access terminal issues the following commands:

- *SuperframePreambleMAC.Activate*
- *SuperframePreambleMAC.Deactivate*
- *OverheadMessages.Activate*
- *OverheadMessages.Deactivate*

11.3.6.2 Indications

This protocol does not register to receive any indications.

11.3.7 Configuration attributes

No configuration attributes are defined for this protocol.

11.3.8 Protocol numeric constants

Protocol numeric constants are listed in Table 262.

Table 262—Protocol numeric constants

Constant	Meaning	Value	Comments
NISPTYPE	Type field for this protocol	See 16.1	
NISPPBasic	Subtype field for this protocol	0x00	
TISPPilotAcq	Time to acquire pilot in access terminal	5 seconds	
TISPSyncAcq	Time to acquire SystemInfo block in access terminal	2 seconds	

11.3.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

11.4 Basic Idle State Protocol

11.4.1 Overview

The Basic Idle State Protocol provides the procedures and messages used by the access terminal and the access network when the access terminal has acquired a network and a connection is not open.

This protocol operates in one of the following four states:

- *Inactive State*: In this state the protocol waits for an Activate command.
- *Sleep State*: In this state the access terminal may shut down part of its subsystems to conserve power. The access terminal does not monitor any Forward Channel, and the access network is not allowed to transmit unicast packets to it. The access terminal and access network remain in the Sleep State when the access terminal performs power down registration.
- *Monitor State*: In this state the access terminal listens for Pages and QuickPages and if necessary, updates the parameters received from the Overhead Messages Protocol. The access network may transmit unicast packets to the access terminal in this state.
- *Access State*: In this state the access terminal sends access preambles to the access network and receives an access grant from the network. This state is not defined for the access network.

Protocol states and events causing the transition between the states are shown in Figure 149 and Figure 150.

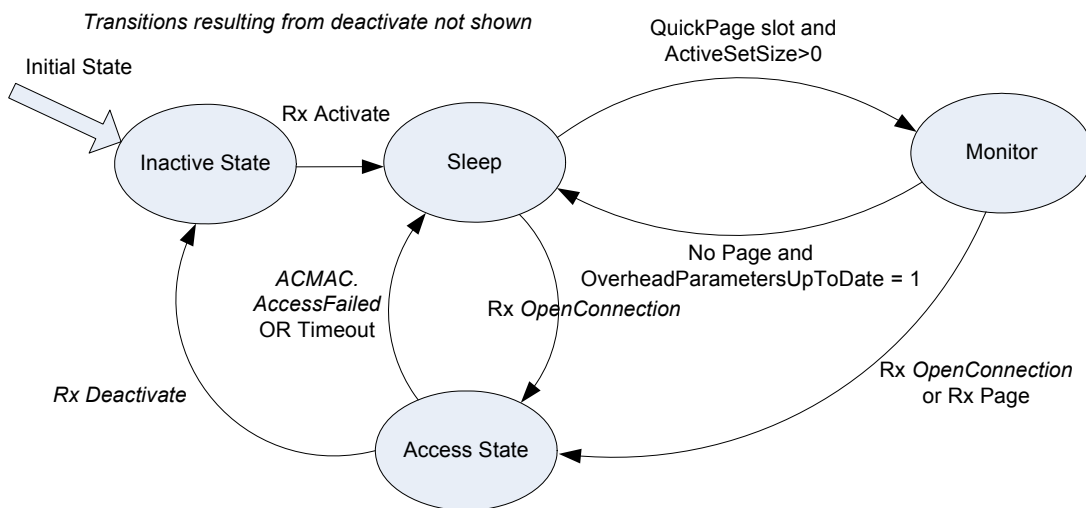


Figure 149—Basic Idle State Protocol (access terminal)

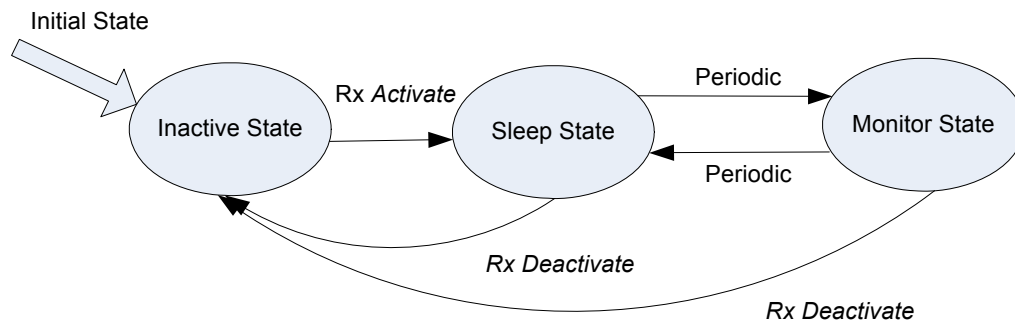


Figure 150—Basic Idle State Protocol State (access network)

This protocol supports periodic network monitoring by the access terminal, allowing for power saving. The following access terminal operations are supported:

- Slotted mode operation, in which the access terminal monitors only selected superframes and sleeps in between. The slotted mode supports staggered operation, where the time interval between the superframes monitored by the terminal increases with time. For the first WakeCount1 sleep instances, the sleep period is PagePeriod1 superframes, and for the next WakeCount2 sleep instances, the sleep period is PagePeriod2 superframes, and for subsequent sleep instances, the sleep period is PagePeriod3 superframes.
- Connection set up: this procedure is initiated by the access terminal, possibly in response to the reception of a page. It consists of the following steps that are performed by the Access Channel MAC Protocol:
 - Access terminal sends an access preamble on R-ACH.
 - Access network sends an access grant on F-SCCH.

There may be multiple InUse instances at the access terminal, and in that case, this protocol at two of the InUse instances performs specific functions:

- Instance corresponding to the session anchor: The Basic Idle State Protocol at this InUse instance determines the times at which the access terminal enters the Monitor State to read pages.
- Instance with a non-null Active Set: This InUse instance performs the operations related to reception of pages, with the time of page reception being determined by the InUse instance corresponding to the session anchor.

If the same InUse instance is Anchor and has a non-null Active Set, then that InUse instance performs both the functions described above.

11.4.2 Primitives

11.4.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*
- *OpenConnection*
- *Close*

11.4.2.2 Return indications

This protocol returns the following indications:

- *ConnectionFailed*

11.4.2.3 Local common data

This protocol defines the following local common data:

- Values of superframe indices where QuickPages may be received

11.4.2.4 Public data

This protocol defines the following public data:

- QuickPageID
- Subtype for this protocol

11.4.3 Procedures and messages for the InConfiguration instance of the protocol

11.4.3.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

11.4.3.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

11.4.3.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

11.4.4 Procedures for the InUse instance

11.4.4.1 Initialization and commit procedures

11.4.4.1.1 Initialization procedures

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the Inactive State.
- The protocol at the access network shall enter the Inactive State.

11.4.4.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

11.4.4.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

11.4.4.2 Command processing

11.4.4.2.1 Activate

When the protocol at the access terminal receives an *Activate* command in the Inactive State:

- The access terminal shall transition to the Sleep State.

When the protocol at the access network receives an *Activate* command in the Inactive State:

- The access network shall transition to the Sleep State.

If the protocol receives this command in any other state the command shall be ignored.

11.4.4.2.2 Deactivate

When the protocol receives a *Deactivate* command in the Inactive State it shall be ignored.

When the protocol receives this command while in any other state:

- The access terminal shall issue an *AccessChannelMAC.Abort* command and transition to the Inactive State.
- The access network shall transition to the Inactive State.

11.4.4.2.3 OpenConnection

When the protocol receives an *OpenConnection* command in the Inactive State or the Access State, the command shall be ignored.

When the protocol receives an *OpenConnection* command in the Sleep State, and PowerDownRegistrationPerformed public data of the Air Link Management Protocol of the session anchor InUse instance is equal to 1, this command shall be ignored.

Otherwise, when the protocol receives this command in the Sleep State:

- The access terminal shall transition to the Access State.
- The access network shall queue the command and execute it when it is in the Monitor State.

When the protocol receives this command in the Monitor State:

- The access terminal shall transition to the Access State.
- The access network shall page the access terminal.

11.4.4.2.4 Close

When the protocol receives a *Close* command in the Inactive State, it shall be ignored.

When the protocol receives a *Close* command while in any other state:

- The access terminal behavior shall be as specified under the procedures for that state.
- The access network behavior is beyond the scope of the specification.

11.4.4.3 General overview of Paging and the Sleep Cycle

11.4.4.3.1 General overview of Paging

Paging is implemented using the F-QPCH and F-DCH physical layer channels. The F-QPCH channel occurs at the beginning of a superframe in superframes with even superframe numbers, and contains a QuickPaging block. If the QuickPaging block in superframe '2n' contains a Quick Page for the access terminal, the full page is sent in superframe '2n + 1'.

The procedures for the reception of a quick page are described in the Superframe Preamble MAC Protocol, while the procedures for the reception of a page are described in the Forward Traffic Channel MAC Protocol.

The Idle State Protocol determines the superframes in which the access terminal may be paged, and activates the Superframe Preamble MAC and the Forward Traffic Channel MAC in the appropriate superframes.

11.4.4.3.2 General overview of Sleep Cycle

This subclause is for informative purposes, and describes which superframes may be used for sending pages by the access network, and receiving pages by the access terminal.

The Basic Idle State Protocol allows the implementation of staggered mode sleep, where the terminal sleep period increases with time, from a sleep period of PagePeriod1 superframes for the first WakeCount1 sleep periods to PagePeriod2 superframes for the next WakeCount2 sleep periods, and finally to PagePeriod3 superframes for the remaining sleep periods (see Figure 151).

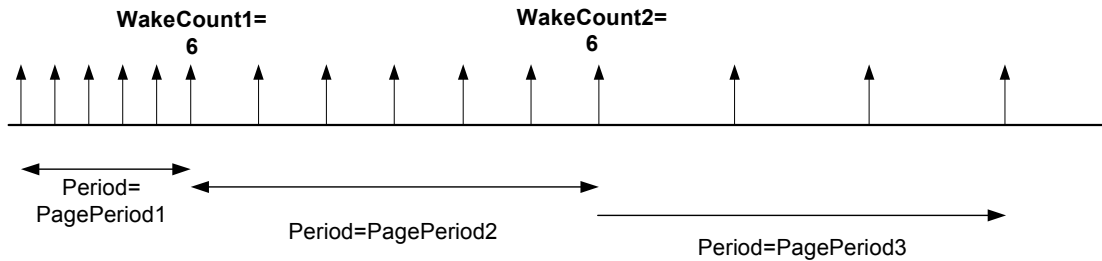


Figure 151—Slotted Mode operation

11.4.4.3.3 General overview of Delay Tolerant Paging

Delay Tolerant Paging is a mechanism that can be used with services that can tolerate a delay in paging. Delay tolerant pages can be deferred to pre-determined intervals where paging is used without quick paging. By not sending quick pages for delay tolerant pages, false wakeups for other users that are monitoring for quick pages can be avoided. An attribute, DelayTolerantPagingCycle, is used to specify the wake-up interval used for delay tolerant pages. DelayTolerantPagingCycle serves as a multiplier of PagePeriod3 such that the access terminal monitors for delay tolerant pages every $\text{PagePeriod3} \times \text{DelayTolerantPagingCycle}$ superframes.

In addition, when the access terminal determines that power savings is not required, it can configure a value of 1 for DelayTolerantPagingCycle to indicate to the access network that use of quick paging can be terminated and thus reduce false wakeups for other users. For example, if the access terminal is connected to an external or high capacity power source, the access terminal can configure a value of 1. When the power source is disconnected, the access terminal can cause the access network to reestablish use of quick paging by configuring DelayTolerantPagingCycle to a value of 0 if pages are not delay tolerant or can increase DelayTolerantPagingCycle if pages are delay tolerant by configuring a value greater than 1.

11.4.4.3.4 General overview of Fast Repaging

Fast Repaging is a mechanism to reduce paging latency in case the access terminal misses a page due to packet error or idle handoff. Consider a page that is sent at time T1, with the next paging opportunity at time $T1 + \text{PagingPeriod}$. In case the page at time T1 is missed by the access terminal, the page could be delivered in the next paging opportunity, resulting in a delay equal to PagingPeriod.

Fast Repaging reduces this delay by introducing a FastRepageInterval attribute, and using the following rules.

- If the access terminal sees a packet error on a paging packet, or misses a paging packet due to idle handoff, then the access terminal listens for a page at time $T1 + \text{FastRepageInterval}$.
- In case the access network does not get a response to a page, the access network may send the page again at time $T1 + \text{FastRepageInterval}$.

11.4.4.4 General procedures

11.4.4.4.1 PagePeriod calculation

The access network and the access terminal shall compute PagePeriod_{*i*} (in units of superframes) according to the Table 263.

Table 263—Computation of PagePeriod_{*i*} from SlotCycle_{*i*}

SlotCycle _{<i>i</i>}	Period _{<i>i</i>}
0x01 to 0x1f	2 ^{SlotCycle_{<i>i</i>}} superframes
0x00 and 0x20 to 0xff	Reserved

11.4.4.4.2 Paging Cycle Offset

R (the paging cycle offset) shall be obtained as follows. Since the paging cycle occurs only once every two superframes, the following procedure is designed to return even values.

- If PreferredPageOffsetEnabled is equal to '0', then *R* is the result of applying the hash function (see 15.1.5) using the following parameters, and multiplying the result by 2:
 - Key = SessionSeed
 - Decorrelate = 6 SessionSeed[11:0]
 - N = Max(PagePeriod1, PagePeriod2, PagePeriod3, 1)
where SessionSeed is given as public data of the Route Control Protocol.
- If PreferredPageOffsetEnabled is equal to '1', then *R* is set to twice the PreferredPageOffset.

11.4.4.4.3 QuickPageID assignment

The parameter QuickPageID is used by the Superframe Preamble MAC Protocol to determine if a quick page is received. The QuickPageID shall be assigned as follows:

- QuickPageID is set to SessionSeed[31:0], where SessionSeed is given as public data of the Route Control Protocol.

11.4.4.4.4 Procedure for Page Time Calculation

The InUse instance corresponding to the session anchor at the access terminal shall compute T_{12} and T_{23} when an *AirLinkManagement.IdleInitiated* indication is received

$$T_{12} = T_c + \text{PagePeriod1} - \left[(T_c + R) \bmod \text{PagePeriod1} \right] + \text{PagePeriod1} \times \left[(\text{WakeCount1}) \right]$$

$$T_{23} = T_{12} + \text{PagePeriod2} - \left[(T_{12} + R) \bmod \text{PagePeriod2} \right] + \text{PagePeriod2} \times \left[(\text{WakeCount2}) \right]$$

where PagePeriod1 and PagePeriod2 are specified in units of superframes, T_c is the current system time and *R* is defined in 11.4.4.4.2.

The InUse instance corresponding to the session anchor at the access terminal shall compute values of *T* that satisfy the following equation, and place them in the public data.

$$[T + R] \bmod \text{PagePeriod} = 0,$$

where *R* is defined in 11.4.4.4.2, and PagePeriod is computed as follows:

$$\text{PagePeriod} = \begin{cases} \text{PagePeriod1}, & T < T_{12} \\ \text{PagePeriod2}, & T_{12} \leq T < T_{23} \\ \text{PagePeriod3}, & \text{Otherwise} \end{cases}$$

11.4.4.4.5 Procedure for ChannelBand selection

The access terminal shall select a ChannelBand from the list of channels in the SectorParameters message. If no channels are listed, the access terminal shall use the ChannelBand it is currently monitoring. If one or more channels are available, the access terminal shall use a hash function (see 15.1.5) to compute an index into the subset of a subset of advertised ChannelBands according to the following procedures.

The access terminal shall create a combined channel list as follows:

- The access terminal shall set the combined channel list to the set of ChannelBands in the channel list.
- If the SupportedChannelBands attribute lists any channels, then the access terminal shall remove from the combined channel list the following ChannelBands:
 - All forward ChannelBands that are not supported by the access terminal as indicated by the SupportedChannelBands attribute.
 - All the forward ChannelBands whose associated reverse ChannelBand is not supported by the access terminal as indicated by the SupportedChannelBands attribute.

The set, S , of ChannelBands is determined as follows:

- The access terminal shall set S to the subset of ChannelBands in the combined channel list for which:
 - N_i is equal to $N_{\max}$, where i is the index of the ChannelBand in the combined channel list, where $N_j = \text{bitcount}(\text{AccessHashingClassMask} [\text{AccessHashingMaskLength}:0] \otimes M_j)$, where M_j is the AccessHashingChannelMask field in the SectorParameters message corresponding to the j^{th} ChannelBand in the combined channel list; $N_{\max}$ is the maximum value of N_k for all k , where k is the index of the ChannelBand in the combined channel list; and $\text{bitcount}(x)$ is the number of '1' bits in the binary representation of x .

The access terminal shall use the following hash function parameters to obtain the index into set S :

- Key = SessionSeed
- Decorrelate = 0
- N = Number of ChannelBands in set S

where SessionSeed is provided as public data by the Route Control Protocol.

The ChannelBand selection procedure described here applies only to access terminal operation for page reception. The access terminal may perform access on other carriers, as controlled by the AccessLoadControl bits transmitted on those carriers. The access terminal may use the AccessLoadControl field of the SectorParameters message to select a carrier to use for access.

11.4.4.5 Inactive State

When the protocol is in the Inactive State it waits for an Activate command.

11.4.4.6 Sleep State

11.4.4.6.1 Access terminal requirements

When the access terminal is in the Sleep State it may stop monitoring the access network by issuing the following commands:

- *SuperframePreambleMAC.Deactivate*
- *ForwardLinkControlSegmentMAC.Deactivate*
- *ForwardTrafficChannelMAC.Deactivate*

The access terminal may shut down processing resources to reduce power consumption.

If PowerDownRegistrationPerformed public data of the Air Link Management Protocol of the session anchor InUse instance is equal to 1, this standard does not specify rules for entering the Monitor State.

If the access terminal has Active Set Size (public data of the Active Set Management Protocol) equal to zero, the access terminal shall not enter the Monitor State. Since there may be more than one InUse instance in Idle State, this requirement provides that only the InUse instance that is currently maintaining radio connectivity enters the Monitor State.

Otherwise (if the two conditions above are not satisfied), the access terminal shall transition out of Sleep state for any one of the following reasons:

- *Update of overhead messages:* To determine the time of transition to the Monitor state, the access terminal may rely on the ChannelInfoExpiryTime, that is public data of the Overhead Messages Protocol. The exact algorithm used by the access terminal to update overhead messages in Idle State is beyond the scope of this standard.
- *To receive quick pages:* The access terminal shall transition to the monitor state at the beginning of superframes where Quick Pages may be sent to the access terminal. The superframe indices for superframes where QuickPages may be sent to the access terminal are public data of the Idle State Protocol.
- *To receive pages:* For a superframe following the superframe where a QuickPage may be sent to the access terminal, the access terminal shall transition to the monitor state to receive a page if
 - A quick page was received in the previous superframe, or
 - F-QPCH was not successfully decoded in the previous superframe, or
 - The attribute DelayTolerantPagingCycle is set to a non-zero value, and the current superframe index satisfies the following equation, where R is defined in 11.4.4.4.2.
$$[T+R] \bmod (\text{PagePeriod} \times \text{DelayTolerantPagingCycle}) = 0$$
- *To receive fast repages:* If the FastRepageInterval attribute is set to a non-zero value, and there is a paging error on the last page read by the access terminal (say in superframe T), or there was a missed page due to idle handoff (say in superframe T), then the access terminal shall transition to the monitor state in superframe T + FastRepageInterval. Here, the paging error event is described by the Forward Traffic Channel MAC Protocol.

If one or more TuneAwaySchedules have been negotiated, the access terminal may continue to operate in Sleep State during the frames when the TuneAwayStatus public data of the Air Link Management Protocol is equal to 1.

11.4.4.6.2 Access network requirements

When the access network is in the Sleep State, it shall not send unicast packets to the access terminal.

If *PowerDownRegistrationPerformed* public data of the Air Link Management Protocol of the session anchor *InUse* instance is equal to 1, then the access network should not enter the monitor state.

The access network enters the monitor state to send a page or a fast repage. The time at which the access network enters the monitor state is outside the scope of the specification, and should be determined by the access network based on the time the access terminal enters the monitor state.³⁶

If one or more *TuneAwaySchedules* have been negotiated, the access network should not transition to Monitor State during frames corresponding to all tune-away schedules. The frames corresponding to tune-away schedules are determined using the procedures defined in the Air Link Management Protocol.

11.4.4.7 Monitor State

The access terminal enters the Monitor State either to receive a Page, Quick Page or other messages from the access network.

When the access network is in the Monitor State, it may send Pages and Quick Pages to the access terminal.

11.4.4.7.1 Access terminal requirements

Upon entering the Monitor State, the access terminal shall:

- Issue *SuperframePreambleMAC.Activate*
- Issue *ForwardTrafficChannelMAC.Activate*
- Issue *ForwardLinkControlSegmentMAC.Activate*
- Issue *OverheadMessages.Activate*

The access terminal shall comply with the following requirements when in the Monitor State:

- If the access terminal receives a *ForwardTrafficChannelMAC.PageReceived* indication, the access terminal shall:
 - Transition to the access state
- If the access terminal has a queued *OpenConnection* command, the access terminal shall transition to the Access State.
- The access terminal may transition to the Access State if it receives a *SuperframePreambleMAC.QuickPageReceived* indication with *QuickPageFormat* = 000 or 001 or a *SuperframePreambleMAC.PageReceived* indication.
- The access terminal may transition to the Access State if the both the following conditions are met:
 - The access terminal receives a *SuperframePreambleMAC.QuickPageReceived* indication with a received *QuickPageFormat* that indicates sufficiently low probability of QuickPage collision. The determination of sufficiently low probability of error may be made by the access terminal using an implementation-dependent algorithm.
 - The access terminal is unable to receive the Page following the QuickPage because of a channel decoding error.

The access terminal may transition to the Sleep State if the Active Set size (public data of the Active Set Management Protocol) is zero, or if all of the following conditions are met:

- *OverheadParametersUpToDate* = 1.

³⁶This is because of possible misalignment between the staggered sleep timelines at the access terminal and access network. In case the connection is closed due to supervision failure, the access network may not know the access terminal settings of T_{12} and T_{23} . In such cases, the access network should use settings for T_{12} and T_{23} that reduce missed pages. The determination of T_{12} and T_{23} at the access network is outside the scope of the specification, and depends on access network implementation

- Transitioning to sleep state will not cause the access terminal to miss a page or Quick Page.

11.4.4.7.2 Access network requirements

The access network shall comply with the following requirements in the Monitor State:

- If the access network has a queued *OpenConnection* command, the access network shall
 - Send the access terminal a QuickPage. Procedures for sending a QuickPage are defined in the Control Channel MAC.
 - In case of delay tolerant paging, which is defined under access terminal procedures, the access network may send a page without sending a quick page.
 - Send the access terminal a Page. Procedures for sending a page are defined in the Forward Traffic Channel MAC. The page packet shall begin transmission in PHY frame index 2 through 7 of the superframe after the superframe where the QuickPage was sent (where the first PHY frame of the superframe has index 0).
 - After the page is sent, transition to the Sleep State.
- If the access network does not have a queued *OpenConnection* command, the access network shall
 - Transition to the Sleep State.

11.4.4.8 Access State

The access terminal and the access network use the Access State to provide a MAC ID to the access terminal.

11.4.4.8.1 Access terminal requirements

The access terminal shall comply with the following requirements:

- Upon entering the Access State the access terminal shall
 - Issue a *SuperframePreambleMAC.Activate* command.
 - Issue a *ForwardLinkControlSegmentMAC.Activate* command.
 - Issue a *ForwardTrafficChannelMAC.Activate* command.
 - Issue a *OverheadMessages.Activate* command.
 - Issue a *AccessChannelMAC.AttemptAccess* command with *AccessType*=0 (InitialAccess).
 - Set a state timer for $T_{IDPATSetup}$ seconds.
- For access in response to a page, the access terminal should accompany the issued *AccessChannelMAC.AttemptAccess* command with the following fields:
 - *AccessPersistenceClass*: This field shall be set no higher than the *AccessPersistenceClass* received with the page (as indicated by the *ForwardTrafficChannelMAC.PageReceived* indication).
 - *PageResponseAccessSequence*: For access in response to a page, this field may be set to the *PageResponseAccessSequence* received with the page (as indicated by the *ForwardTrafficChannelMAC.PageReceived* indication).
- For access initiated by the access terminal, the access terminal should accompany the issued *AccessChannelMAC.AttemptAccess* command with the following fields:
 - *StreamNN*: This field shall be set to the identifier of one of the streams with unsent data in the buffer.
- If the state timer expires or the protocol receives an *AccessChannelMAC.AccessFailed* indication or a *Close* command, the access terminal shall
 - Issue an *AccessChannelMAC.Abort* command.

- Return a ConnectionFailed indication.
- Transition to the Sleep State.

11.4.4.8.2 Access network requirements

The Access State is not applicable to the Idle State Protocol at the Access Network.

11.4.5 Message formats for the InUse instance

No messages are defined for the InUse instance of this protocol.

11.4.6 Interface to other protocols

11.4.6.1 Commands

This protocol issues the following commands:

- *OverheadMessages.Activate*
- *SuperframePreambleMAC.Activate*
- *SuperframePreambleMAC.Deactivate*
- *ForwardTrafficChannelMAC.Activate*
- *AccessChannelMAC.AttemptAccess*
- *AccessChannelMAC.Abort*
- *ForwardLinkControlSegmentMAC.Activate*
- *ForwardLinkControlSegmentMAC.Deactivate*

11.4.6.2 Indications

This protocol registers to receive the following indications:

- *AirLinkManagement.IdleInitiated*
- *AccessChannelMAC.AccessFailed*
- *SuperframePreambleMAC.PageReceived*
- *SuperframePreambleMAC.QuickPageReceived*
- *ForwardTrafficChannelMAC.PageReceived*

11.4.7 Configuration attributes

11.4.7.1 Simple attributes

The access terminal and the access network shall use as defaults the values that are listed in ***bold italics*** (see Table 264).

Table 264—Configurable values

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	FastRepageInterval	Soft Allowed/Static	<i>0</i>	FastRepage disabled
			0x01–0xff	The interval at which the access network pages the access terminal when the access network receives no response to the page. The unit for this field shall be superframes. This field shall not take odd values.
0x0001	DelayTolerantPagingCycle	Soft Allowed/Static	<i>0</i>	Delay tolerant paging is not enabled.
			0x01–0xff	The interval, in units of PagePeriod3, when the access terminal monitors pages even if it did not receive a QuickPage.
0x0002	AccessHashingClassMask	Soft Allowed/Static	<i>0x0000</i>	Access terminal and access network will hash to channels with any access hashing class.
			0x0001–0xffff	Access terminal and access network will hash to channels with designated access hashing classes.

11.4.7.2 Complex attributes

The following complex attributes and default values are defined.

11.4.7.2.1 Preferred Paging attribute

This attribute may be used to override the default value of the PageOffset value R (see Table 265).

This attribute has AttributeID 0x8000.

Table 265—Preferred Paging attribute

Field	Length (bits)	Default
PreferredPageOffsetEnabled	1	0
PreferredPageOffset	14	0
Reserved	1	0

PreferredPageOffsetEnabled

This field shall be set to '1' if the PreferredPageOffset field is tused; otherwise, the sender shall set this field to '0'.

PreferredPageOffset If PreferredPageOffsetEnabled is set to '1', this field shall specify the superframe in which the access terminal transitions out of the Sleep State (see 11.4.4.6) in order to monitor the Control Channel. This field shall be ignored if PreferredPageOffsetEnabled is set to '0'.

Reserved This field shall be ignored by the receiver.

Commit	Soft allowed

Scope	Dynamic

11.4.7.2.2 SlottedMode attribute

This attribute has AttributeID 0x8001 (see Table 266).

Table 266—SlottedMode attribute

Field	Length (bits)	Default
SlotCycle1	8	0x9
SlotCycle2	8	0x9
SlotCycle3	8	0x9
WakeCount1	8	0x0
WakeCount2	8	0x0

SlotCycle1 The sender shall set this field to SlotCycle1.

SlotCycle2 The sender shall set this field to SlotCycle2. SlotCycle2 shall be greater than or equal to SlotCycle1.

SlotCycle3 The sender shall set this field to SlotCycle3. SlotCycle3 shall be greater than or equal to SlotCycle2.

WakeCount1 The sender shall set this field to WakeCount1.

WakeCount2 The sender shall set this field to WakeCount2. WakeCount2 shall be greater or equal to than WakeCount1.

Commit	Soft allowed

Scope	Static

11.4.7.2.3 SupportedChannelBands attribute

The access terminal uses this attribute to convey to the access network the ChannelBands supported by the access terminal (see Table 267).

This attribute has AttributeID 0x8002.

Table 267—SupportedChannelBands attribute

Field	Length (bits)	Default
BandClassCount	8	0x00

Table 267—SupportedChannelBands attribute (continued)

Field	Length (bits)	Default
BandClassCount occurrences of the following record{		
BandClass	8	N/A
BandSubClassCount	8	N/A
BandSubClassCount occurrences of the following record{		
BandSubClass	8	N/A
}}		

BandClassCount	The access terminal shall set this field to the number of occurrences of the BandClass field in this complex value.
BandClass	The access terminal shall set this field to the band class supported by the access terminal.
BandSubClassCount	The access terminal shall set this field to the number of band sub-classes supported by the access terminal in this band class.
BandSubClass	The access terminal shall set this field to the band sub-class supported by the access terminal.

Commit	Soft allowed

Scope	Static

11.4.8 Protocol numeric constants

Table 268 lists the protocol numeric constants.

Table 268—Protocol numeric constants

Constant	Meaning	Value	Comments
N _{IDPType}	Type field for this protocol	See 16.1	
N _{IDPBasic}	Subtype field for this protocol	0x00	
T _{IDPATSetup}	Timeout	2s	

11.4.9 Session state information

The Session State Information record (see 15.2.6) consists of the parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

11.5 Basic Connected State Protocol

11.5.1 Overview

The Basic Connected State Protocol provides procedures and messages used by the access terminal and the access network while a connection is open.

This protocol can be in one of states:

- *Inactive State*: In this state the protocol waits for an Activate command.
- *BindATI State*: In this state, the access terminal has an assigned MAC ID. This state provides a binding of the assigned MAC ID to the access terminal's ATI.
- *Open State*: In this state the access terminal has an assigned MAC ID that is bound to the access terminal's ATI.
- *Semi-Connected State*: In this state, the access terminal enters a power saving mode while maintaining a MAC ID.
- *Close State*: This state is associated only with the access network. In this state the access network waits for connection resources to be safely released.

Figure 152 and Figure 153 show the state transition diagrams at the access terminal and the access network respectively.

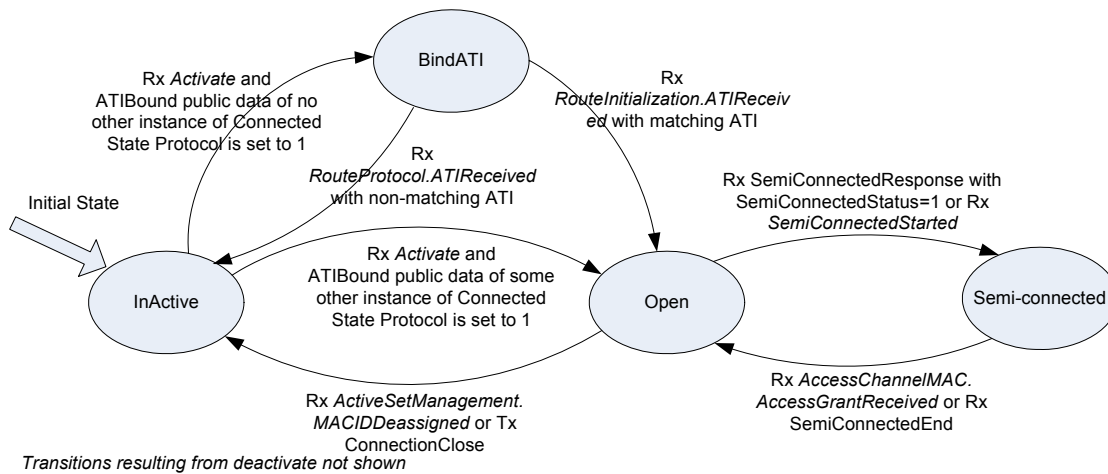


Figure 152—Basic Connected State Protocol State Diagram (access terminal)

Transitions resulting from deactivate not shown

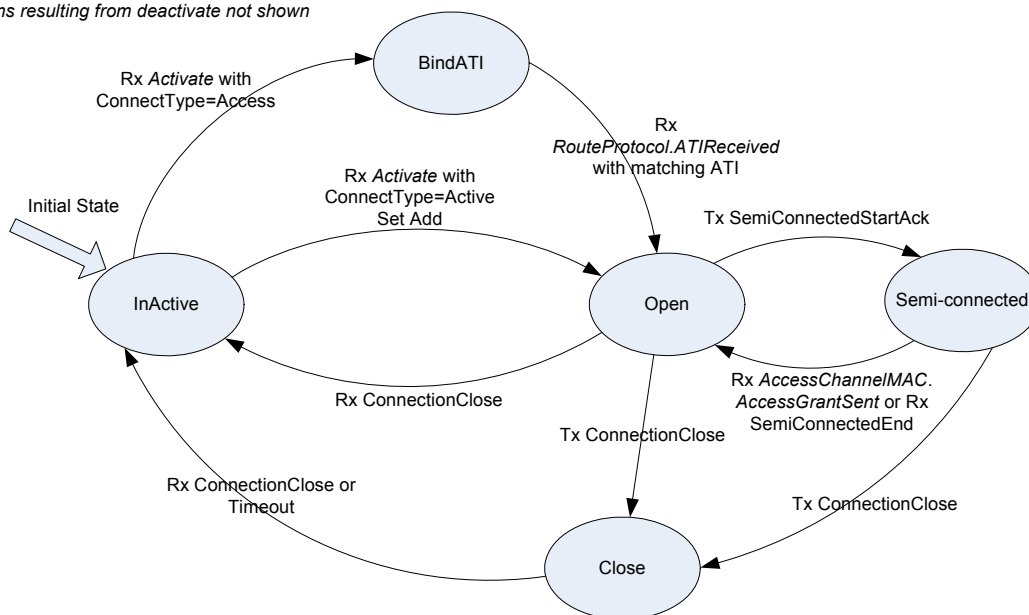


Figure 153—Basic Connected State Protocol State Diagram (access network)

This protocol provides the following functions:

- After access, binding the access terminal identifier to the MAC ID assigned in the access grant.
 - The access terminal sends a packet on the R-ODCH. This packet contains an access terminal identifier in the Route Initialization Protocol header.
 - Access network sends a response packet on the F-DCH. The response packet contains an access terminal identifier in the Route Protocol header.
- Managing the close of a connection.
- Managing semi-connected state operation: Semi-connected state is a power saving state where the access terminal retains an Active Set, but does not transmit reverse control channels. The terminal exits semi-connected state by making an access attempt with a reserved access sequence.
- Managing selected interlace operation.

11.5.2 Primitives

11.5.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*
- *CloseConnection34F*
- *ExitSemiConnectedState*

11.5.2.2 Return indications

This protocol returns the following indications:

- *ConnectionClosed*

- *SemiConnectedStarted*
- *SemiConnectedEnded*
- *SemiConnectedRejected*
- *SemiConnectedRequested*

11.5.2.3 Local common data

This protocol defines the following local common data:

- *AccessBasedHandoffRequired*
- *SemiConnectedAckTimerActive*
- *SemiConnectedServingSector*

11.5.2.4 Public data

This protocol defines the following public data:

- Subtype for this protocol
- *ATIBound* (AT only)
- *SelectedInterlaceMode*
- *SelectedInterlaceSet*
- State of this protocol

11.5.3 Procedures and messages for the InConfiguration instance of the protocol

11.5.3.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

11.5.3.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

11.5.3.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

11.5.4 Procedures for the InUse instance

11.5.4.1 Initialization and commit procedures

11.5.4.1.1 Initialization procedures

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the Inactive State.
- The protocol at the access network shall enter the Inactive State.

11.5.4.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

11.5.4.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

11.5.4.2 Command processing

11.5.4.2.1 Activate

When the protocol at the access terminal receives an *Activate* command in the Inactive State, the access terminal shall

- Issue a *ReverseTrafficChannelMAC.Activate* command.
- Issue a *ReverseControlChannelMAC.Activate* command.
- If ATIBound public data of some other instance of Connected State Protocol is set to 1, the access terminal shall set ATIBound to 1 and transition to the Open State.
- Otherwise, the access terminal shall set ATIBound to 0 transition to the BindATI State.

When the protocol at the access network receives an *Activate* command in the Inactive State, the access network shall

- Issue a *ReverseTrafficChannelMAC.Activate* command.
- Issue a *ReverseControlChannelMAC.Activate* command.
- If the command is accompanied by *ConnectType* = *ActiveSetAdd*, the access network shall transition to the Open State.
- Otherwise, the access network shall transition to the BindATI State.

When the protocol at the access terminal or access network receives this command in any other state, it shall be ignored.

11.5.4.2.2 Deactivate

When the protocol receives a *Deactivate* command in the Inactive State it shall ignore the command.

When the protocol receives a *Deactivate* command in any other state, it shall perform the cleanup procedures defined in 11.5.4.4.1.3.

11.5.4.2.3 CloseConnection

When the protocol receives a *CloseConnection* command in the Close State or the InActive State, the command shall be ignored.

When the protocol receives this command in the Open State, Semi-Connected State or the BindATI State:

- Access terminal shall send a *ConnectionClose* message to the access network and perform the cleanup procedures defined in 11.5.4.4.1.3.
- Access network shall send a *ConnectionClose* message to the access terminal and transition to the Close State.

11.5.4.2.4 ExitSemiConnectedState

The access terminal and access network shall ignore this command in all states except the Semi-Connected State.

The procedures for processing this command in Semi-Connected State are described under the procedures for that state.

11.5.4.3 BindATI State**11.5.4.3.1 Access terminal requirements**

The access terminal shall comply with the following requirements.

- Upon entering the BindATI State the access terminal shall start a state timer for T_{CSPBind} seconds.
- If the state timer expires the access terminal shall execute the cleanup procedures given in 11.5.4.3.1.2.
- If the access terminal receives a *ConnectedState.ConnectionClosed* indication from another InUse instance, it shall execute the cleanup procedures given in 11.5.4.3.1.2.

11.5.4.3.1.1 Processing the RouteProtocol.ATIReceived indication

The access terminal shall process this indication according to the following rules.

The access terminal shall declare the indication valid if the ATIdentifier that accompanies the indication matches the ATIdentifier last transmitted by the access terminal in the Route Protocol header (public data of Route Protocol).

- If the indication is not valid, the access terminal shall execute the cleanup procedures given in 11.5.4.3.1.2.
- If the access terminal receives a valid indication with RouteStatus equal to 0x1, it shall execute the cleanup procedures given in 11.5.4.3.1.2.
- If the access terminal receives a valid indication with RouteStatus equal to 0x0 it shall perform the following:
 - Reset the state timer.
 - Set ATIBound to '1'.
 - Transition to Open State.

11.5.4.3.1.2 Cleanup procedures for the BindATI State

The access terminal shall do the following when this procedure is invoked:

- Return a *ConnectionClosed* indication.
- Transition to the InActive State.

11.5.4.3.2 Access network requirements

Upon entering the BindATI State, the access network should send a packet with the Route Protocol header included.

- If the access network does not accept the connection (e.g., when it returns an error code in the Route Protocol header), this protocol shall
 - Return a *ConnectionClosed* indication.
 - Transition to the InActive State.

- If the access network accepts the connection, and receives a MAC packet that does not include an extended Route header, this protocol shall transition to the Open State.

11.5.4.4 Open State

In the Open State, the access terminal and the access network maintain a connection.

11.5.4.4.1 Access terminal requirements

11.5.4.4.1.1 General requirements

The access terminal shall comply with the following requirements when in the Open State:

- The access terminal may request *SelectedInterlace* operation for power saving.
- The access terminal may transition to Semi-connected state for power saving.
- The access terminal may close the connection by sending a *ConnectionClose* message. If the access terminal sends a *ConnectionClose* message, it shall execute the cleanup procedures defined in 11.5.4.4.1.3.
- If the access terminal receives a *ConnectedState.ConnectionClosed* indication from another *InUse* instance, the access terminal shall execute the cleanup procedures defined in 11.5.4.4.1.3.
- If the access terminal receives a *ConnectionClose* message, it shall send a *ConnectionClose* message with *CloseReason* set to “Close Reply” and execute the cleanup procedures defined in 11.5.4.4.1.3.

11.5.4.4.1.2 Channel measurement procedures

The access network may obtain channel measurement reports from the access terminal by sending a *ChannelMeasurementReportRequest* message.

If an access terminal receives a *ChannelMeasurementReportRequest* message, the access terminal may respond with a *ChannelMeasurementReport* message. Channel measurements are based on the F-CPICH.

11.5.4.4.1.3 Cleanup procedures

If the access terminal executes cleanup procedures it shall

- Return a *ConnectionClosed* indication.
- Set *ATIBound* to 0.
- Transition to the Inactive State.

11.5.4.4.2 Access network requirements

11.5.4.4.2.1 General requirements

The access network shall comply with the following requirements when in the Open State:

- If the access network receives a *ConnectionClose* message, it shall consider the connection closed, and it should execute the cleanup procedures defined in 11.5.4.4.2.2.
- If the access network requires closing the connection, it shall transmit a *ConnectionClose* message, and transition to the Close State.

11.5.4.4.2.2 Cleanup procedures

When the access network performs cleanup procedures it shall:

- Return a *ConnectionClosed* indication.

- Transition to the Inactive State.

11.5.4.4.3 Procedures for Semi-Connected Operation

This subclause describes procedures governing transitions from Open State to Semi-Connected State. The access terminal or access network shall not enter Semi-Connected State if *SemiConnectedModeSupported* attribute is zero.

11.5.4.4.3.1 Access terminal procedures for InUse instance initiating transition

The access terminal shall not send a *SemiConnectedRequest* message if any of the conditions is satisfied:

- Public data *SemiConnectedAckTimerActive* is equal to 1.
- An *ActiveSetManagement.MACResourceUpdated* indication has been received from some InUse instance in the last T_{CSPSCR} seconds.

To transition to the Semi-Connected state, the InUse instance corresponding to the sector that the access terminal shall be monitoring in Semi-Connected State shall perform the following:

- Generate a *SemiConnectedRequested* indication including a *SemiConnectedServingSector* flag set to the sector that the access terminal shall be monitoring in Semi-Connected State.
- Send a *SemiConnectedRequest* message including a *SemiConnectedServingSector* field set to the sector that the access terminal shall be monitoring in Semi-Connected State.
- Start a *SemiConnectedAck* timer for T_{CSPSCA} seconds.
- Set *SemiConnectedAckTimerActive* to 1.
- Set *AccessBasedHandoffRequired* to 1.
- Set *SemiConnectedServingSector* to the corresponding field in the *SemiConnectedRequest* message.

If the *SemiConnectedAck* timer expires, the access terminal shall

- Transition to the Semi-Connected State. A transition to Semi-Connected State is required in this case because the access network may have entered Semi-Connected State.
- Generate a *SemiConnectedEnd* message with the *SemiConnectedServingSector* set to the *SemiConnectedServingSector* field in the corresponding *SemiConnectedRequest* message.
- Set *SemiConnectedAckTimerActive* to 0 and disable the *SemiConnected* timer.

If the access terminal receives a valid *SemiConnectedResponse* message with *SemiConnectedStatus* equal to '1', (where validity is defined according to the message sequence field, as defined in 11), the access terminal shall

- Disable the *SemiConnected* timer.
- Set public data *SemiConnectedAckTimerActive* to 0.
- Set public data *AccessBasedHandoffRequired* to 1.
- Transition to the Semi-Connected State.

If the access terminal receives a valid *SemiConnectedResponse* message with *SemiConnectedStatus* equal to '0', (where validity is defined according to the message sequence field, as defined in 11) message, the access terminal shall

- Disable the *SemiConnected* timer.
- Generate a *SemiConnectedRejected* indication including a *SemiConnectedServingSector* flag set to the *SemiConnectedServingSector* field in the corresponding *SemiConnectedRequest* message.
- Send a *SemiConnectedEnd* message to the sector indicated by the *SemiConnectedServingSector* field in the corresponding *SemiConnectedRequest* message.

- Set public data *SemiConnectedAckTimerActive* to 0.
- Set public data *SemiConnectedServingSector* to NULL.
- Set public data *AccessBasedHandoffRequired* to 0.

11.5.4.4.3.2 Access terminal procedures for InUse instances not initiating transition

If the access terminal receives a *SemiConnectedRequested* indication from some other InUse instance, it shall send a *SemiConnectedRequest* message with *SemiConnectedServingSector* set to the *SemiConnectedServingSector* field received with the indication.

If the access terminal receives a *SemiConnectedRejected* indication from some other InUse instance, it shall send a *SemiConnectedEnd* message with *SemiConnectedServingSector* set to the *SemiConnectedServingSector* field received with the indication.

If the access terminal receives a *ConnectedState.SemiConnectedStarted* indication from some other InUse instance, it shall perform the following:

- Transition to the Semi-Connected State.

11.5.4.4.3.3 Access network procedures

If the access network receives a *SemiConnectedRequest* message, and the *SemiConnectedServingSector* included in the message is part of the Active Set assigned by the access network, the access network should respond with a *SemiConnectedResponse* message within T_{CSPSCA} seconds. If the access network sets *SemiConnectedStatus* to '1', the access network shall transition to the Semi-Connected State.

If the access network receives a *SemiConnectedRequest* message, and the *SemiConnectedServingSector* included in the message is not part of the Active Set assigned by the access network, the access network should not respond to the message, and may transition to the Semi-Connected State.

11.5.4.4.4 SelectedInterlace Operation procedures

The access terminal and access network may operate in one of two modes: *SelectedInterlaceOn* or *SelectedInterlaceOff*.

- *SelectedInterlaceOn* mode: In this mode, the access network sends certain SCCH blocks to the access terminal only on a set of interlaces called the *SelectedInterlaceSet*. Details may be found in 4.
- *SelectedInterlaceOff* mode: In this mode, no restrictions are placed on the access network and access terminal.

The access terminal may request transition to the *SelectedInterlaceOn* mode if the *SelectedInterlaceSupported* attribute is equal to '1'. The access terminal shall not request transition to the *SelectedInterlaceOn* mode if the *SelectedInterlaceSupported* attribute is equal to '0'.

The *SelectedInterlace* mode may be used to reduce power consumption at the access terminal, by allowing the access terminal to turn off its transmitter and receiver during some frames. The *SelectedInterlace* mode does not always allow the access terminal to turn off its receiver and transmitter in all interlaces outside the *SelectedInterlaceSet*. The only requirement on the access network in *SelectedInterlace* mode is that assignments to the access terminal should be sent only on the *SelectedInterlaceSet*. The access terminal may be required to transmit on other interlaces for the following purposes:

- CQI Transmissions
- Reception of power control commands from the serving sector

- Transmission on the R-ODCH or R-CDCH (corresponding to assignments received on the SelectedInterlaceSet)
- Other requirements imposed by the Lower MAC Sublayer

11.5.4.4.1 State transitions for selected interlace operation

The access network shall enter the SelectedInterlaceOn mode after

- Sending a SelectedInterlaceAssignment message with SelectedInterlaceEnabled set to '1', if the pilot identified in the message corresponds to the current serving sector.

The access terminal shall enter the SelectedInterlaceOn mode after

- Receiving a SelectedInterlaceAssignment message with SelectedInterlaceEnabled set to '1', if the pilot identified in the message corresponds to the current serving sector.

The access terminal shall enter the SelectedInterlaceOff mode after

- Upon protocol initialization.
- Receiving a SelectedInterlacesAssignment message with SelectedInterlacesEnabled equal to '0'.
- When the desired serving sector is not the same as the serving sector, i.e., when the access terminal is in the process of switching from one sector to another.
- When the forward and reverse link serving sectors are different.
- When the access terminal sends a SemiConnectedRequest message.
- When the Forward Link Serving Sector and Reverse Link Serving Sector are different.

The access network shall enter the SelectedInterlaceOff mode after

- Upon protocol initialization.
- Receiving a SelectedInterlaceAck message with the pilot identified in the message corresponds to the current serving sector.
- The serving sector for the access terminal changes.
- When the forward and reverse link serving sectors are different.
- When the access network sends a SemiConnectedResponse message.

If the access terminal receives a SelectedInterlaceAssignment message with SelectedInterlacesEnabled equal to '0', the access terminal shall respond with a SelectedInterlaceAck message.

To change the selected interlace assignment to an access terminal, the access network should first disable selected interlace mode, and then send a SelectedInterlaceAssignment message.

11.5.4.4.2 Procedures in Selected Interlace States

On entering the SelectedInterlaceOn state:

- The access terminal and access network shall set public data SelectedInterlaceMode to '1'.
- Place the most recent SelectedInterlaceAssignment message in the public data.

On entering the SelectedInterlaceOff state:

- The access terminal and access network shall set public data SelectedInterlaceMode to '0'.

11.5.4.5 Semi-Connected State

In Semi-Connected State, the access terminal periodically monitors F-SCCH for assignments. If the access terminal receives an assignment (indicated by *ForwardLinkControlSegmentMAC.SemiConnectedEndReceived*), it performs access procedures and transitions to the Open State. The access terminal may also initiate access procedures at any time to enter the Open State. The access terminal and access network cannot exchange any Unicast data in Semi-Connected State.

The Lower MAC Sublayer performs the following functions relating to Semi-Connected State:

- When the Connected State Protocol is in Semi-Connected State, the Lower MAC Sublayer
 - Looks out for the special LAB corresponding to Semi-connected exit at *SemiConnectedWake* times.
 - Does not transmit any RL channels.
 - MAC protocols may be deactivated outside *SemiConnectedWake* times.
- When *AccessBasedHandoffRequired* is 1, the Lower MAC Sublayer
 - Sets $RLSS = FLSS$.
 - Performs handoff using access sequences.

11.5.4.5.1 Access terminal procedures

Upon entering this state, the access terminal shall return a *ConnectedState.SemiConnectedStarted* indication, and upon exiting this state, the access terminal shall return a *ConnectedState.SemiConnectedEnded* indication.

The access terminal shall monitor the Forward Control Segment from the semi-connected serving sector (as indicated in the *SemiConnectedRequest* message) in frames (denoted by PHY frame Index) that simultaneously satisfy the both the following inequalities for some integer j .

$$PHY\ frame\ Index \geq j \times SemiConnectedPeriod + SemiConnectedOffset$$

$$PHY\ frame\ Index < j \times SemiConnectedPeriod + SemiConnectedOffset + SemiConnectedWakeDuration$$

The access terminal may shut down processing resources at other times to reduce power consumption.

If the access terminal receives a *ForwardLinkControlSegmentMAC.SemiConnectedEndReceived* indication, or an *ExitSemiConnectedState* command, the access terminal shall issue an *AccessChannelMAC.AttemptAccess* command with *AccessType* = 1.

If the access terminal wishes to perform handoff in this state, the access terminal shall issue an *AccessChannelMAC.AttemptAccess* command with *AccessType* = 1. The Access Channel MAC is responsible for directing the access attempt to the appropriate serving sector.

If the access terminal receives an *AccessChannelMAC.AccessGrantReceived* indication, it shall perform the following:

- Send a *SemiConnectedEnd* message.
- Transition to the Open State.

If the access terminal receives a *ConnectedState.SemiConnectedEnded* indication from some other InUse instance, it shall perform the following:

- Send a *SemiConnectedEnd* message.

- Transition to the Open State.

11.5.4.5.2 Access network procedures

In SemiConnected State, the access network

- May stop monitoring the reverse control channel.
- Shall not send Unicast F-DCH packets to the access terminal.
- May cause the access terminal to access by sending a signal on F-SCCH during frames when the access terminal is known to be monitoring F-SCCH (as specified in access terminal procedures for SemiConnected State).

If the access network receives a SemiConnectedEnd message, it shall

- Send a SemiConnectedResponse message with the SemiConnectedStatus field to 0.
- Transition to the Open State.

11.5.4.6 Close State

The Close State is associated only with the access network. In this state the access network waits for a replying ConnectionClose message from the access terminal or for the expiration of the “CSP Close Timer” defined below.

Upon entering this state, the access network shall set a “CSP Close Timer” for an implementation dependent duration. The duration of this timer may be based on the KeepAlive timer defined by the Route Control Protocol. If the access network receives a ConnectionClose message in this state, or if the timer expires, it shall execute the cleanup procedures defined in 11.5.4.4.2.2, it may close all connection-related resources assigned to the access terminal, and it should transition to the Inactive State.

11.5.5 Message formats for the InUse instance

11.5.5.1 ConnectionClose

The access terminal and the access network send the ConnectionClose message to close the connection (see Table 269).

Table 269—ConnectionClose message

Field	Length (bits)
MessageID	8
CloseReason	3
RouteCounterIncluded	1
RouteCounter	15 ^a
Padding	Variable

^aThis field may be absent (see field description for conditions).

MessageID The sender shall set this field to 0x00.

CloseReason The sender shall set this field to reflect the close reason, as shown in Table 270.

Table 270—Encoding of the CloseReason Field

Field value	Description
'000'	Normal Close; Reason Unspecified
'001'	Close Reply
'010'	Connection Error
'011'	Normal close requested by access terminal because the connection was opened for registration.
All other values are reserved	

- RouteCounterIncluded

This field shall be set to '1' if the sender is the access terminal. This field shall be set to '0' otherwise.
- RouteCounter

This field shall be included if RouteCounterIncluded is equal to '1', and shall be omitted otherwise. The access terminal shall set this field to RouteCounter public data of the Route Control Protocol.
- Padding

The length of this field shall be such that the entire message is octet-aligned. The sender shall set this field to zero. The receiver shall ignore this field.

Channels	FTC	RTC	RLP	Best effort
Addressing	Unicast		AuthTag	Required if key is available

11.5.5.2 SelectedInterlaceRequest

This message shall be sent by the access terminal to request a selected interlace mode with a particular sector (see Table 271).

Table 271—SelectedInterlaceRequest message

Field	Length (bits)
MessageID	8
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
PartialInterlaceRequested	2
SelectedInterlacesBitmap	8
Reserved	3

- MessageID
- This field shall be set to 0x01.

PilotID	This field shall be set to the PilotID of the sector to which this message is directed. The access network shall ignore this message if the PilotID does not match the PilotID of the sector that received the message.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.
PartialInterlaceRequested	The access terminal shall set this field to '00' to indicate partial interlace is not requested, to '01' to request 1/2 interlace, '10' for 1/3 interlace, and '11' for 1/4 interlace.
SelectedInterlacesBitmap	The i^{th} entry of this field shall be set to 1 if interlace i is in the set of SelectedInterlaces requested by the access terminal, and shall be set to zero otherwise. The rules for setting this field in case partial interlaces are requested is the same as for the corresponding field of the SelectedInterlaceAssignment message.
Reserved	The receiver shall ignore this field.

Channels	RTC
Addressing	Unicast

RLP	Best effort
AuthTag	Required when key is available

11.5.5.3 SelectedInterlaceAssignment

This message shall be sent by the access network to assign a selected interlace mode to an access terminal (see Table 272).

Table 272—SelectedInterlaceAssignment

Field	Length (bits)
MessageID	8
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
SelectedInterlaceEnabled	1
PartialInterlaceAssignment	2 ^a
SelectedInterlacesBitmap	8 ^a
Padding	Variable

^aThis field may be absent (see field description for conditions).

MessageID	This field shall be set to 0x02.
PilotID	This field shall be set to the PilotID of the sector to which this message is directed.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the PilotID to which this message refers. This field shall be omitted otherwise.
SelectedInterlaceEnabled	This field shall be set to 1 if the access terminal is required to operate in SelectedInterlaceOn mode, and is set to 0 if the access terminal is required to operate in SelectedInterlaceOff mode.
PartialInterlaceAssignment	The access network shall omit this field if SelectedInterlaceEnabled is equal to '0'. Otherwise, the access network shall set this field to '00' to indicate partial interlace is not assigned, to '01' to assign 1/2 interlace, '10' for 1/3 interlace, and '11' for 1/4 interlace.
SelectedInterlacesBitmap	The access network shall omit this field if SelectedInterlaceEnabled is equal to '0'. Otherwise, the i^{th} entry of this field shall be set to 1 if interlace i is in the set of SelectedInterlaces, and shall be set to zero otherwise. If DuplexingMode = FDD, in case partial 1/2 interlace is assigned, an access terminal with even MAC ID will use frames of the form $i + 16n$, and an access terminal with odd MAC ID will use frames of the form $8 + i + 16n$, where $n = 0, 1, 2, \dots$. If DuplexingMode = TDD44, in case partial 1/2 interlace is assigned, an access terminal with even MAC ID will use frames of the form $i + 8n$, and an access terminal with odd MAC ID will use frames of the form $4 + i + 8n$, where $n = 0, 1, 2, \dots$. If DuplexingMode = TDD63, in case partial 1/2 interlace is assigned, an access terminal with even MAC ID will use frames of the form $i + 12n$, and an access terminal with odd MAC ID will use frames of the form $6 + i + 12n$, where $n = 0, 1, 2, \dots$. If DuplexingMode = FDD, in case partial 1/3 interlace is assigned, the access terminal will use frames of the form $i + (\text{MAC ID mod } 3) \times 8 + 24n$. If DuplexingMode = TDD44, in case partial 1/3 interlace is assigned, the access terminal will use frames of the form $i + (\text{MAC ID mod } 3) \times 4 + 12n$. If DuplexingMode = TDD63, in case partial 1/3 interlace is assigned, the access terminal will use frames of the form $i + (\text{MAC ID mod } 3) \times 6 + 18n$. If DuplexingMode = FDD, in case partial 1/4 interlace is assigned, the access terminal will use frames of the form $i + (\text{MAC ID mod } 4) \times 8 + 32n$. In the TDD mode, in case partial 1/4 interlace is assigned, the access terminal will use frames of the form $i + (\text{MAC ID mod } 4) \times 4 + 16n$. If DuplexingMode = TDD63, in case partial 1/4 interlace is assigned, the access terminal will use frames of the form $i + (\text{MAC ID mod } 4) \times 6 + 24n$.
Padding	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. The sender shall set this field to zero. The receiver shall ignore this field.

Channels	FTC

RLP	Best effort

Addressing	Unicast

AuthTag	Required when key is available

11.5.5.4 SelectedInterlaceAck

This message shall be sent by the access terminal to acknowledge transition to SelectedInterlaceOff state (see Table 273).

Table 273—SelectedInterlaceAck message

Field	Length (bits)
MessageID	8
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
Reserved	5

MessageID This field shall be set to 0x03.

PilotID This field shall be set to the PilotID of the sector to which this message is directed. The access network shall ignore this message if the PilotID does not match the PilotID of the sector that received the message.

ChannelBandIncluded This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.

ChannelBand This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.

Reserved The receiver shall ignore this field.

Channels	RTC

RLP	Best effort

Addressing	Unicast

AuthTag	Required when key is available

11.5.5.5 SemiConnectedRequest

This message shall be sent by the access terminal to the access network to request Semi-Connected Operation (see Table 274).

Table 274—SemiConnectedRequest message

Field	Length (bits)
MessageID	8
MessageSequence	16
SemiConnectedServingSector	10
SemiConnectedPeriod	8
SemiConnectedOffsetIncluded	1
SemiConnectedOffset	8 ^a
SemiConnectedWakeDuration	4
Reserved	1

^aThis field may be absent (see field description for conditions).

- MessageID** The access terminal shall set this field to 0x04.
- MessageSequence** The access terminal shall set this field to the sequence number of this message. The sequence number of this message is 1 more than the sequence number of the last SemiConnectedRequest message (modulo 65536) sent by this access terminal. If this is the first SemiConnectedRequest message sent by the access terminal, it shall set this field to 0x00.
- SemiConnectedServingSector** This field shall be set to the PilotID of the serving sector that the access terminal monitors during periodic wakeup in Semi-Connected state.
- SemiConnectedPeriod** This field determines the interval at which the access terminal monitors the SemiConnectedServingSector for an indication of pending forward link traffic. This field is in units of PHY frames, and shall not be set to '0'.
- SemiConnectedOffsetIncluded** This field shall determine if the SemiConnectedOffset is included in this message. The access terminal should include this field to suggest a value of SemiConnectedOffset to the access network.
- SemiConnectedOffset** This field shall be included if SemiConnectedOffsetIncluded is '1', and shall be omitted otherwise. This field determines the offset of frames during which the access terminal monitors the SemiConnectedServingSector for an indication of pending forward link traffic. This field is in units of PHY frames, and shall take values less than SemiConnectedPeriod.
- SemiConnectedWakeDuration** This field determines the number of frames the access terminal monitors at each

monitoring instance. This field is in units of PHY frames, and shall not be set to '0'.

Reserved The receiver shall ignore this field.

Channels	RTC	RLP	Best effort
Addressing	Unicast	AuthTag	Required when key is available

11.5.5.6 SemiConnectedResponse

This message shall be sent by the access network to control transitions between Open State and Semi-Connected State (see Table 275).

Table 275—SemiConnectedResponse message

Field	Length (bits)
MessageID	8
MessageSequence	16
SemiConnectedStatus	1
SemiConnectedFailureCode	3 ^a
The following record shall be included if SemiConnectedStatus is '1'{	
SemiConnectedPeriodDurationOverride	1
SemiConnectedPeriod	8
SemiConnectedWakeDuration	4 ^a
SemiConnectedOffsetOverride	1
SemiConnectedOffset	8 ^a
}	
Padding	Variable

^aThis field may be absent (see field description for conditions).

- MessageID This field shall be set to 0x05.
- MessageSequence The access network shall set this field to the corresponding field of the last received SemiConnectedRequest message sent to this access terminal.
- SemiConnectedStatus The access network shall set this field to one to indicate successful transition to Semi-Connected State. The access network shall set this field to zero to indicate transition to Connected State.

SemiConnectedFailureCode

This field shall be omitted if SemiConnectedStatus is equal to '1'. Otherwise, this field shall be set to a failure code value as specified in Table 276.

Table 276—SemiConnectedFailureCode

Reason	Value
Semi-Connected Mode not supported	001
Failure due to non-empty buffer	010
General failure	011
Response to SemiConnectedEnd message	100
Reserved	All other values

SemiConnectedPeriodDurationOverride

This field shall be omitted if SemiConnectedStatus is equal to '0', and included otherwise. If included, this field shall be set to '1' if the SemiConnectedPeriod and SemiConnectedWakeDuration to be used by the access terminal is different from the one included in the corresponding SemiConnectedRequest message, and to '0' otherwise.

SemiConnectedPeriod

This field shall be omitted if the SemiConnectedPeriodDurationOverride field is omitted or equal to '0'. Otherwise, this field determines the interval at which the access terminal monitors the SemiConnectedServingSector for an indication of pending forward link traffic. This field is in units of PHY frames, and shall not be set to zero.

SemiConnectedWakeDuration

This field shall be omitted if the SemiConnectedPeriodDurationOverride field is omitted or equal to '0'. Otherwise, this field determines the number of frames the access terminal monitors at each monitoring instance. This field is in units of PHY frames, and shall not be set to zero.

SemiConnectedOffsetOverride

This field shall be omitted if SemiConnectedStatus is equal to '0', and included otherwise. If included, this field shall be set to '1' if the corresponding SemiConnectedRequest message did not include a SemiConnectedOffset field.

SemiConnectedOffset

This field shall be omitted if the SemiConnectedOffsetOverride field is omitted or equal to '1'. Otherwise, this field determines the offset of frames during which the access terminal monitors the SemiConnectedServingSector for an indication of pending forward link traffic. This field is in units of PHY frames, and shall take values less than SemiConnectedPeriod.

Padding

The length of this field shall be such that the message is octet aligned. The sender shall set this field to zero. The receiver shall ignore this field.

Channels	FTC
Addressing	Unicast

RLP	Best effort
AuthTag	Required when key is available

11.5.5.7 SemiConnectedEnd

This message shall be sent by the access terminal to inform the access network of a transition from Semi-Connected State to Open State (see Table 277).

Table 277—SemiConnectedEnd message

Field	Length (bits)
MessageID	8
MessageSequence	16

MessageID This field shall be set to 0x06.

MessageSequence The access terminal shall set this field to the corresponding field of the last received SemiConnectedResponse message.

Channels	RTC

RLP	Best effort

Addressing	Unicast

AuthTag	Required when key is available

11.5.5.8 ChannelMeasurementReportRequest

The access network sends this message to request a ChannelMeasurementReport from one or more access terminals (see Table 278).

Table 278—ChannelMeasurementReportRequest message

Field	Length (bits)
MessageID	8
PilotPN	12
CarrierID	2
StartPHYFrame	44
NumChannels	3
MeasurementsPerMessage	8
NumMeasurementsRequested	8
Reserved	3

MessageID This field shall be set to 0x0b.

PilotPN This field shall be set to the PilotPN of the sector requesting the measurement report.

CarrierID This field shall be set to the carrier on which the measurements are requested.

StartPHYFrame This field shall be set to the PHY frame number of the PHYFrame where access terminals are required to begin measurements.

- NumChannels

This field shall be set to the number of channels to be measured by the access terminal. Each measured channel corresponds to a different transmit antenna at the sector being measured.
- MeasurementsPerMessage

This field shall determine the number of measurements (in terms of PHY frames measured) to be included in one ChannelMeasurementReport message.
- NumMeasurementsRequested

This field shall determine the total number of measurements to be made by the access terminal.
- Reserved

The receiver shall ignore this field.

Channels	FTC		RLP	Reliable
Addressing	Broadcast	Unicast	AuthTag	Required when key is available

11.5.5.9 ChannelMeasurementReport

The access terminal sends this message to report channel measurements (see Table 279).

Table 279—ChannelMeasurementReport message

Field	Length (bits)
MessageID	8
PilotID	12
CarrierID	2
StartPHYFrameNumber	44
MeasurementInterval	8
NumMeasurements	8
NumMeasurements instances of the following record{	
NumChannels	3
NumChannels instances of the following record{	
NumTaps	3
NumTaps instances of the following record{	
TapOffset	5
RealGain	8
ImagGain	8
}}	
Padding	Variable

MessageID	This field shall be set to 0x0c.
PilotID	This field shall be set to the PilotID of the sector for which the measurement was performed.
CarrierID	This field shall be set to the carrier on which the measurements are performed.
StartPHYFrameNumber	This field shall be set to the PHY frame number of the PHY frame where access terminal made the first measurement reported in this message.
MeasurementInterval	This field shall determine the number of PHY frames between measurements made by the access terminal.
NumMeasurements	This field shall be set to the number of measurements included in this message. Each measurement corresponds to a different PHY frame.
NumChannels	This field determines the number of channels measured by the access terminal. The access terminal shall set this field to equal to the NumChannels field in the received ChannelMeasurementReportRequest message.
NumTaps	This field shall be set to the number of taps being reported.
TapOffset	This field shall be set to a offset for which the channel was measured.
RealGain	This field shall be set to the real component of the measured channel gain on the corresponding TapOffset.
ImagGain	This field shall be set to the imaginary component of the measured channel gain on the corresponding TapOffset.
Padding	The length of this field shall be such that the entire message is octet-aligned. The sender shall set this field to zero. The receiver shall ignore this field.

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

11.5.6 Interface to other protocols

11.5.6.1 Commands

This protocol issues the following commands:

- *ReverseTrafficChannelMAC.Activate*
- *ReverseControlChannelMAC.Activate*

11.5.6.2 Indications

This protocol registers to receive the following indications:

- *RouteProtocol.ATIReceived* (AT only)
- *ActiveSetManagement.MACResourceUpdated*

- *ForwardLinkControlSegmentMAC.SemiConnectedEndReceived*
- *AccessChannelMAC.AccessGrantReceived*
- *ConnectedState.SemiConnectedStarted* (from all InUse instances)
- *ConnectedState.SemiConnectedEnded* (from all InUse instances)
- *ConnectedState.SemiConnectedRejected* (from all InUse instances)
- *ConnectedState.SemiConnectedRequested* (from all InUse instances)

11.5.7 Configuration attributes

The following configuration attributes are defined for this protocol.

11.5.7.1 Simple attributes

The access network and the access terminal shall use the default values that are typed in ***bold italics*** (see Table 280).

Table 280—Configurable simple attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	SemiConnectedModeSupported	Soft Allowed/Static	<i>0x00</i>	Semi-Connected mode not supported.
			0x01	Semi-Connected mode is supported.
			All other values	Reserved
0x0001	SelectedInterlaceModeSupported	Soft Allowed/Static	<i>0x00</i>	SelectedInterlace mode not supported.
			0x01	SelectedInterlace mode is supported.
			All other values	Reserved.

11.5.7.2 Complex attributes

This protocol does not define any complex attributes.

11.5.8 Protocol numeric constants

The protocol numeric constants are listed in Table 281.

Table 281—Protocol numeric constants

Constant	Meaning	Value	Comments
N _{CSPType}	Type field for this protocol	See 16.1	
N _{CSPBasic}	Subtype field for this protocol	0x0000	
T _{CSPSCA}	Time for which the access terminal waits for a SemiConnectedAck message	500 ms	
T _{CSPBind}	Maximum access terminal time in the BindATI State	0.2 seconds	

Table 281—Protocol numeric constants (continued)

Constant	Meaning	Value	Comments
T_{CSPSCR}	Time duration after change in Active Set when the access terminal does not initiate a transition to Semi-Connected State.	2.0 seconds	

11.5.9 Session State Information

The Session State Information record (see 15.2.6) consists of the parameter records.

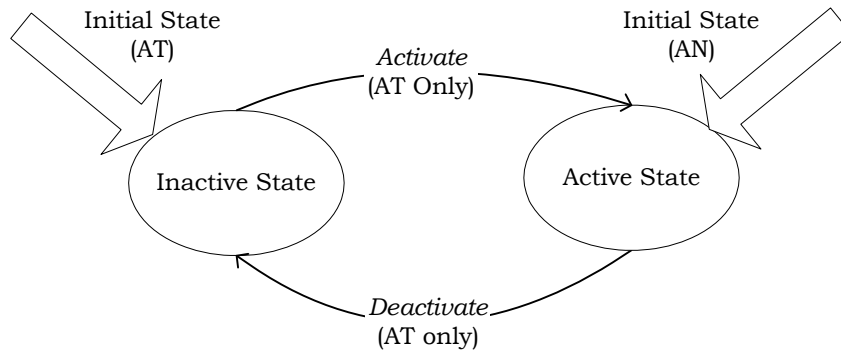
11.6 Overhead Messages Protocol

11.6.1 Overview

The Overhead Messages Protocol is responsible for the transmission, reception and supervision of the SystemInfo block, the QuickChannelInfo block, the ExtendedChannelInfo message and the SectorParameters message. The SystemInfo and QuickChannelInfo blocks are broadcast by the access network directly over the Superframe Preamble MAC Protocol. The ExtendedChannelInfo and SectorParameters messages are broadcast using the Signaling Protocol.

This protocol can be in one of two states:

- *Inactive State*: In this state, the protocol waits for an *Activate* command. This state corresponds only to the access terminal and occurs when the access terminal has not acquired an access network or is not required to receive overhead messages.
- *Active State*: In this state the access network transmits and the access terminal receives overhead messages.

**Figure 154—Overhead Messages Protocol State Diagram**

11.6.2 Primitives

11.6.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

11.6.2.2 Return indications

This protocol returns the following indications:

- *SupervisionFailed*
- *QuickChannelInfoUpdated*
- *OverheadMessages.SystemInfoReceived*

11.6.2.3 Local common data

This protocol does not define any local common data.

11.6.2.4 Public data

This protocol defines the following public data:

- An *OverheadParameterList* that shall contain for each (PilotPN, ChannelBand) pair in the active set, the following entries:
 - Received *SystemInfo* block, and computed *SBCCHScramblingSeed*
 - Received *AcqInfo* block
 - Received *QuickChannelInfo* block
 - Received *ExtendedChannelInfo* message with associated *ChannelInfoExpiryTime*
 - Currently valid *SectorParameters* messages indexed by the sector's (PilotID, ChannelBand)
 - *SectorChannelInfoUpToDate*
- The following Boolean entries:
 - *OverheadParametersUpToDate*
 - *ChannelInfoUpToDate*
 - *SectorParametersUpToDate*
- Subtype for this protocol

11.6.3 Procedures and messages for the InConfiguration instance of the protocol

11.6.3.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

11.6.3.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

11.6.3.3 Message formats

The InConfiguration instance of this protocol does not define any messages.

11.6.4 Procedures for the InUse instance

11.6.4.1 Initialization and commit procedures

11.6.4.1.1 Initialization procedures

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the Inactive State.
- The protocol at the access network shall enter the Active State.

11.6.4.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

11.6.4.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

11.6.4.2 Extensibility requirements

Further revisions of the access network may add new overhead messages.

The access terminal shall discard overhead messages with a MessageID field it does not recognize.

Further revisions of the access network may add new fields to existing overhead messages. These fields shall be added to the end of the message, prior to the Reserved field, if such a field is defined.

The access terminal shall ignore fields it does not recognize.

11.6.4.3 Command processing

The access network shall ignore all commands.

11.6.4.3.1 Activate

If this protocol receives an *Activate* command in the Inactive State:

- The access terminal shall transition to the Active State.
- The access network shall ignore it.

If this protocol receives the command in the Active State, it shall be ignored.

11.6.4.3.2 Deactivate

If this protocol receives a *Deactivate* command in the Inactive State, it shall be ignored.

If this protocol receives the command in the Active State:

- Access terminal shall transition to the Inactive State.
- Access network shall ignore it.

11.6.4.4 Inactive State

This state corresponds only to the access terminal and occurs when the access terminal has not acquired an access network or is not required to receive overhead messages. In this state, the protocol waits for an *Activate* command.

11.6.4.5 Active State

11.6.4.5.1 Access network requirements

If the access network is ready to provide service, it shall broadcast the SystemInfo block, QuickChannelInfo block, ExtendedChannelInfo message and SectorParameters message as specified below. The SystemInfo block, QuickChannelInfo block, ExtendedChannelInfo message and the SectorParameters message shall be public data of the Overhead Messages Protocol.

11.6.4.5.1.1 Procedure for transmission of the AcqInfo Block

The AcqInfo block shall be transmitted every superframe. The AcqInfo block shall be carried by the Superframe Preamble MAC Protocol over the TDM Pilot 3 physical channel, and shall not pass through the Signaling Protocol.

11.6.4.5.1.2 Procedure for transmission of the SystemInfo Block

The SystemInfo block shall be transmitted every $N_{\text{PBCCH\_Period}}$ superframes. The SystemInfo block shall be carried by the Superframe Preamble MAC Protocol over the F-PBCCH physical channel, and shall not pass through the Signaling Protocol. $N_{\text{PBCCH\_Period}}$ is defined in the Lower MAC Sublayer Protocol.

11.6.4.5.1.3 Procedure for transmission of the QuickChannelInfo Block

The QuickChannelInfo block shall be transmitted in every superframe with an odd superframe index. The QuickChannelInfo block shall be carried by the Superframe Preamble MAC Protocol, over the F-SBCCH physical channel, and shall not pass through the Signaling Protocol. The information carried in the QuickChannelInfo block transmitted in superframe $m = 2k + 1$ describes the structure of

- All PHY frames except the first two PHY frames of superframe $2k + 1$
- All PHY frames of superframe $2k + 2$
- The first two PHY frames in superframe $2k + 3$

The access network shall scramble the CRC on F-SBCCH with a SystemInfoSignature, as described in 11.6.4.5.1.6.

11.6.4.5.1.4 Procedure for transmission of the ExtendedChannelInfo message

The ExtendedChannelInfo message shall be broadcast over the Forward Traffic Channel MAC. The message shall begin transmission in superframes with superframe numbers of the form $1 + j \times N_{\text{OMPECI}}$, where j is an integer. Further, this message shall begin transmission PHY frames 2 through 7 of superframe $1 + j \times N_{\text{OMPECI}}$, i.e., it shall not begin transmission in the first two PHY frames of this superframe, and shall not begin transmission in PHY frame 8 of this superframe or later. Depending on HARQ at the Physical Layer, the ExtendedChannelInfo message may be delivered in one superframe, or in a set of consecutive superframes.

If transmission of an ExtendedChannelInfo message begins in superframe $2k - 1$, the ExtendedChannelInfo message shall describe the structure of

- All PHY frames except the first two PHY frames of superframe $2k + 1$.

- All PHY frames of superframes $2k + 2$ through $2k + N_{\text{OMPECI}}$.
- The first two PHY frames in superframe $2k + N_{\text{OMPECI}} + 1$.

11.6.4.5.1.5 Procedure for transmission of the SectorParameters message

The access network shall send a SectorParameters message over the Forward Traffic Channel MAC in superframe numbers of the form $1 + j \times N_{\text{OMPSectorParameters}}$, where j is an integer. Further, this message shall begin transmission PHY frames 2 through 7 of superframe $1 + j \times N_{\text{OMPSectorParameters}}$, i.e., it shall not begin transmission in the first two PHY frames of this superframe, and shall not begin transmission in PHY frame 8 of this superframe or later. If an ExtendedChannelInfo and SectorParameters message are transmitted in the same superframe, the access network shall set the SectorParametersSignature field of the ExtendedChannelInfo message to the SectorParametersSignature field of SectorParameters message. Otherwise, The access network shall set the SectorParametersSignature field of the ExtendedChannelInfo message to the SectorParametersSignature field of the next SectorParameters message.

The access network may include the AccessLoadControl field in the SectorParameters for a carrier other than the current carrier advertised in the channel list if the information is available. The SectorParametersSignature need not change when other channel's information has changed.

11.6.4.5.1.6 Procedures for validity timers

The access network shall change the QuickChannelInfo block and the ECISignature fields in accordance with the ChannelInfoValidityIndex field advertised in the ExtendedChannelInfo message. If an ExtendedChannelInfo message begins transmission in superframe k , and given the ChannelInfoValidityIndex field in the ExtendedChannelInfo message transmitted in superframe k , then the access network shall:

- Set v as the smallest integer greater than k that is of the form $(\text{PilotID} + j \times 256 \times 2^{\text{ChannelInfoValidityIndex}}) \times N_{\text{OMPECI}}$, for some integer j .
- Set the same ECISignature field for all subsequent ExtendedChannelInfo messages until superframe v .
- The earliest time the access network may change the ECISignature field is in superframe $v + 1$.
- Set the same QuickChannelInfo block for superframes k to v .
- The earliest time the access network may change the QuickChannelInfo block is in superframe $v + 1$.
- If the access network changes the QuickChannelInfo block, or the ECISignature fields at the validity timer, then, before the new parameters take effect, the access network shall send the updated parameters to all connected state users, possibly by tunneling the information to these users using the EncapsulatedQuickChannelInfo and EncapsulatedExtendedChannelInfo messages.
- If the access network does not change the QuickChannelInfo block, or the ECISignature fields at the validity timer, the access network should send the updated parameters to all connected state users before the new parameters take effect, possibly by tunneling the information to these users using the EncapsulatedQuickChannelInfo and EncapsulatedExtendedChannelInfo messages.

A pictorial representation of changing overhead parameters is given in Figure 155.

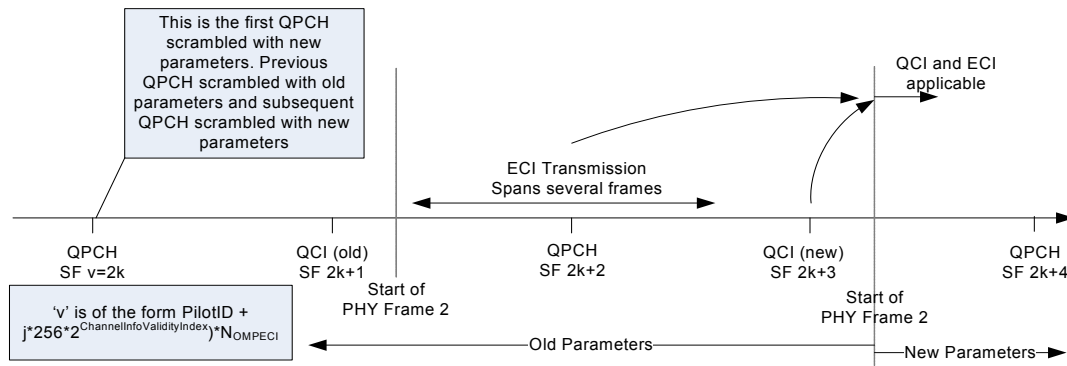


Figure 155—Overhead message timeline

11.6.4.5.1.7 Procedures for generating Scrambling Seeds

11.6.4.5.1.7.1 General Hash function for generating Scrambling Seeds

This procedure takes as input an arbitrary length bit-string I , and returns a 20-bit hash quantity H .

- Compute $0 \leq n < 64$ and m with length $(I) = m \times 64 - n$
- Initialize H to zero
- Set string J to n zeros followed by string I , and set $i = 0$
- While $i < m$, repeat steps e) and f)
- $H = H \oplus f_{\text{PHY-HASH}}(J(i \times 64 : i \times 64 + 63))$
- $i = i + 1$

Here $f_{\text{PHY-HASH}}$ is the common PHY hash function defined in the Physical Layer, and $J(a:b)$ denotes a substring of J , starting a position 'a' and ending at position 'b' (both inclusive).

11.6.4.5.1.7.2 Computation of Scrambling Seed for F-SBCCH

The F-SBCCH channel is scrambled by the Physical Layer using a seed calculated from the SystemInfo block that is transmitted next by the same sector on F-PBCCH. To provide time-variation of the seed, this protocol generates four 20-bit seeds, denoted by $\text{SBCCHScramblingSeed}(i)$, with $i = 0, 1, 2, 3$. $\text{SBCCHScramblingSeed}(i)$ is calculated by invoking the procedure in 11.6.4.5.1.7.1 with the $\text{SystemInfoBlock}:i$ as input, where ':' denotes concatenation, 'i' is '00', '01', '10', or '11', and the SystemTimeLSB field of the SystemInfoBlock is replaced by zeros.

For a F-SBCCH transmission in a given SuperframeIndex, the access network shall use the seed with index $i = \text{floor}((\text{SuperframeIndex} \bmod 8)/2)$.

11.6.4.5.1.7.3 Computation of Scrambling Seed for F-QPCH

The F-QPCH channel is scrambled by the Physical Layer using a seed calculated from the ECISignature field (24 bits) of the ExtendedChannelInfo message and the QuickChannelInfo block transmitted by the same sector.

To provide time-variation of the seed, this protocol generates four 20-bit seeds, denoted by $QPCHScramblingSeed(j)$, with $j = 0, 1, 2, 3$. For a F-QPCCCH transmission in a given SuperframeIndex the access network shall use the seed with index $j = \text{floor}((\text{SuperframeIndex} \bmod 8)/2)$.

These seeds are computed by performing the following steps, for $j = 0, 1, 2, 3$.

- a) Compute $QCISignature(j)$ by invoking the procedure in 11.6.4.5.1.7.1 with QuickChannelInfo:j as input, where ':' denotes concatenation and 'j' is '00', '01', '10', or '11'.
- b) Compute $ECISignature(j) = f_{PHY-HASH}(ECISignature:i)$, where ':' denotes concatenation and 'j' is '00', '01', '10', or '11'.
- c) Compute $QPCHScramblingSeed(i) = QCISignature(j) \oplus ECISignature(j)$.

If ECISignature or QuickChannelInfo block change with time (as governed by rules for validity timers), the QPCHScramblingSeed used in Superframe 2k shall be computed using

- The QuickChannelInfo block transmitted in Superframe $2k + 1$, and
- The most recent ExtendedChannelInfo message that began transmission in or before Superframe $2k + 1$.

11.6.4.5.2 Access terminal requirements

In the Active State, the access terminal shall invoke the procedure for determining if the OverheadMessages are up-to-date, as specified in 11.6.4.5.2.8. When in the Active State, the access terminal shall perform supervision on the QuickChannelInfo, ExtendedChannelInfo and the SectorParameters messages as specified in 11.6.4.5.2.8.

When the access terminal receives a ExtendedChannelInfo message from a sector, it shall perform the procedures in 11.6.4.5.2.1.

When the access terminal receives a SectorParameters message from a sector, it shall perform the procedures in 11.6.4.5.2.5.

11.6.4.5.2.1 Procedure for processing the AcqInfo Block

The access terminal shall place the received AcqInfo block, indexed by PilotID and ChannelBand, in the public data.

11.6.4.5.2.2 Procedure for processing the SystemInfo Block

If the public data does not contain any SystemInfo block, the access terminal shall generate a *OverheadMessages.SystemInfoReceived* indication. The access terminal shall place the received SystemInfo block, indexed by PilotID and ChannelBand, in the public data.

11.6.4.5.2.3 Processing the QuickChannelInfo Block and EncapsulatedQuickChannelInfo message

If a QuickChannelInfo block is received in superframe n, then at the end of the second PHY frame of superframe n, the access terminal shall:

- If the received QuickChannelInfo block differs from the stored block in the public data, the access terminal shall generate a *QuickChannelInfoUpdated* indication.
- Store the block, indexed by PilotID and ChannelBand, in the public data.

11.6.4.5.2.4 Processing the ExtendedChannelInfo message

When the access terminal receives a ExtendedChannelInfo message from a sector, it shall perform the following:

- The access terminal shall determine the superframe number n when the access network started ExtendedChannelInfo transmission.
- At the end of the second PHY frame (Frame 1) of superframe $n + 2$, the access terminal shall perform the following operations:
 - Store the ExtendedChannelInfo message, indexed by PilotID and ChannelBand in the public data.
 - Set the ChannelInfoExpiryTime to the end of the second PHY frame of superframe $v + 3$, where v is the smallest integer greater than n that is of the form $(\text{PilotID} + j \times 256 \times 2^{\text{ChannelInfoValidityIndex}}) \times N_{\text{OMPECI}}$, for some integer j . Here, ChannelInfoValidityIndex is a field of the received ExtendedChannelInfo message.
- When the access terminal adds a ExtendedChannelInfo message to the public data, it shall process the stored SectorParameters messages according to the following rule:
 - If the public data contains a SectorParameters message with the same PilotID and ChannelBand as the sector that transmitted the ExtendedChannelInfo message, the access terminal shall compare the SectorParametersSignature in the ExtendedChannelInfo message with the SectorParametersSignature in the stored SectorParameters message. If the signatures do not match, the access terminal shall purge the SectorParameters message from the public data.

11.6.4.5.2.5 Processing a QuickPage block

Based on the Physical Layer scrambling on the QuickPageBlock, the access terminal may extend the ChannelInfoValidityIndex time upon error-free reception of a QuickPageBlock. If a QuickPageBlock is received without error in superframe n , the access terminal may set the ChannelInfoExpiryTime to the end of the second PHY frame of superframe $v + 3$. Here v is the smallest integer greater than n that is of the form $(\text{PilotID} + k \times 256 \times 2^{\text{ChannelInfoValidityIndex}}) \times N_{\text{OMPECI}}$ for some integer k , and ChannelInfoValidityIndex is a field of the last received ExtendedChannelInfo message.

11.6.4.5.2.6 Processing the SectorParameters message

When the access terminal receives a SectorParameters message, it shall perform the following:

- If the public data contains a SectorParameters message with the same SectorID as the received message, the access terminal shall compare the SectorParametersSignature of the received message with the SectorParametersSignature in the stored SectorParameters message. If the signatures do not match, the access terminal shall
 - Replace the SectorParameters message in the public data with the received SectorParameters message.
 - If the sector is a member of the Active Set, return a *SectorParametersUpdated* indication.
- If the public data does not contain a SectorParameters message with the SectorID of the received message, the access terminal shall
 - Add the received SectorParameters message to the public data.
 - If the sector is a member of the Active Set, return a *SectorParametersUpdated* indication.
- If necessary, the access terminal may delete old SectorParameters messages corresponding to sectors not in the Active Set.

11.6.4.5.2.7 Processing the EncapsulatedQuickChannelInfo and EncapsulatedExtendedChannelInfo messages

The access terminal shall buffer these messages and process them at the beginning of superframe number EffectiveSuperframeNumber.

11.6.4.5.2.8 Procedure for checking if parameters are up-to-date

Using this set of procedures, the access terminal determines ChannelInfoUpToDate and SectorParametersUpToDate as follows:

For a given sector, SectorChannelInfoUpToDate shall be set to '1' if the following conditions are true, and shall be set to '0' otherwise.

- An ExtendedChannelInfo message and QuickChannelInfo block for this member of the Active Set is available in the public data.
- The ChannelInfoExpiryTime for this sector is greater than or equal to the current time.

ChannelInfoUpToDate shall be set to '1' if SectorChannelInfoUpToDate=1 for the following members of the Active Set: RLSS, FLSS, DRLSS, DFLSS (as indicated by the public data of the Reverse Control Channel MAC Protocol). This field shall be set to '0' otherwise.

SectorParametersUpToDate shall be set to '1' if all of the following conditions are satisfied for all members of the Active Set, and shall be set to '0' otherwise.

- A SectorParameters message with the same PilotID as this member of the Active Set is available in the public data.
- An ExtendedChannelInfo block for this member of the Active Set is available in the public data.
- The SectorParametersSignature in the last received ExtendedChannelInfo block is the same as the SectorParametersSignature in the stored SectorParameters message.

OverheadParametersUpToDate shall be set to the logical “and” of SectorParametersUpToDate and ChannelInfoUpToDate. The OverheadParametersUpToDate field is used by the Idle State Protocol.

The access terminal shall not transmit on the reverse link to a sector for which one or more of the following conditions is satisfied:

- An ExtendedChannelInfo message and QuickChannelInfo block for this member of the Active Set is available in the public data.
- The ChannelInfoExpiryTime for this sector is less than the current time.

11.6.4.5.2.9 AN groups

The access terminal should maintain AN groups based on received SectorParameters messages and Active Sets in the public data of Active Set Management Protocol. If there are multiple InUse instances at the access terminal, the information about AN groups shall be maintained in a database that is common across all instances of this protocol. In addition, there shall be an AN Group associated with each InUse instance. This associated AN group shall include the pilots in the Active Set of this InUse instance.

The AN group helps pilot management at the Active Set Management Protocol, by helping the access terminal determine which pilots belong to existing InUse instances, and thereby helping the access terminal minimize the creation of new InUse instances to add pilots to the Active Set.

The access terminal may use an implementation dependent sizing of the AN group database, and should purge pilots from the database using implementation-dependent algorithms. The exact algorithm to maintain AN groups is implementation dependent.

11.6.4.5.3 Supervision procedures

11.6.4.5.3.1 Supervision of ChannelInfo

When the Air Link Management Protocol is in Connected State, the access terminal shall use the following procedure to supervise the QuickChannelInfo block and ExtendedChannelInfo message:

- The access terminal shall set a ChannelInfo supervision timer for $T_{\text{OMP CISupervision}}$.
- If ChannelInfoUpToDate becomes '1' while the timer is active, the access terminal shall disable the timer. If ChannelInfoUpToDate becomes '0' while the timer is inactive, the access terminal shall start the timer.
- If the timer expires, the access terminal shall return a *SupervisionFailed* indication and disable the timer.

11.6.4.5.3.2 Supervision of SectorParameters message

The access terminal shall use the following procedure to supervise the SectorParameters message:

- The access terminal shall set a SectorParameters supervision timer for $T_{\text{OMP SPSupervision}}$.
- If SectorParametersUpToDate becomes '1' while the timer is active, the access terminal shall disable the timer. If SectorParametersUpToDate becomes '0' while the timer is inactive, the access terminal shall start the timer.
- If the timer expires, the access terminal shall return a *SupervisionFailed* indication and disable the timer.

11.6.5 Message and block formats

In the interpretation of these messages, the symbol 'n' is used to denote the value of a bit field. For example, the field CPLength is assigned two bits, and the parameter CPLength takes values $N_{\text{FFT}} \times (1 + n)/16$, where n is the 2-bit field that takes the value 0, 1, 2, or 3.

11.6.5.1 AcqInfo block

This protocol shall place the following AcqInfo block in the public data for transmission by the Superframe Preamble MAC. The structure of the AcqInfo block is a function of the IsSynchronous field of the block (see Table 282 and Table 283).

Table 282—AcqInfo information block in Asynchronous mode

Field	Number of bits	Description
IsSynchronous	1	This bit is set to 0.
FDDHalfDuplexTDDPartition	1	This bit is set to 1 if half-duplex operation is supported, and is set to 0 otherwise.
SuperframeIndexLSBs	4	This field contains the four LSBs of the SuperframeIndex, starting from the most significant bit to the least significant bit.
Reserved	3	These bits are ignored by the receiver.

Table 283—AcqInfo Information Block in Synchronous mode

Field	Number of bits	Description
IsSynchronous	1	This bit is set to 1.
EnablePreambleFrequencyReuse	1	This bit is set depending on whether or not frequency reuse on the preamble is enabled.
FDDHalfDuplexTDDPartition	1	When TDDEnabled is '0', the bit FDDHalfDuplexTDDPartition is carried, and is set to 1 if half-duplex operation is supported, and is set to 0 otherwise. When TDDEnabled is '1', the bit TDDMode is carried, DuplexingMode is TDD44 if this bit is equal to '0', DuplexingMode is TDD63 if this bit is equal to '1'.
TDDEnabled	1	If this bit is equal to '1' then DuplexingMode is TDD44 or TDD63, and if this bit is equal to 0 then DuplexingMode is FDD.
Reserved	5	These bits are ignored by the receiver.

11.6.5.2 SystemInfo block

The SystemInfo block shall be transmitted directly by the Superframe Preamble MAC Protocol over the F-PBCC channel, and shall not pass through other Layers. The SystemInfo block shall have the following format (see Table 284).

Table 284—SystemInfo block

Field	Length (bits)
IPSI0	4
IPSI1	4
IPSI2	4
IPSI3	4
SystemTimeLSB	9
TotalNumSubcarriers	3
NumGuardSubcarriers	6
NumShortChannelIDs	2
FLReservedInterlaces	4
NumFLReservedSubzones	4
Reserved	4

IPSI0-3 These fields shall be set to IPSIs supported by this sector. Additional IPSIs may be included in the SectorParameters message.

SystemTimeLSB	This field shall be set to the nine lower bits of the superframe number at the time the SystemInfo block starts transmission.
TotalNumSubcarriers	This parameter shall take the value $2^{(7+n)}$, where n is the value of the 3 bit field. This field shall not be set to a value of 5 or above.
NumGuardSubcarriers	This field shall be set to the number of GuardSubcarriers, in units of 16. The number of guard subcarriers shall be less than TotalNumSubcarriers/2.
NumShortChannelIDs	This field shall be set to the number of ChannelBands over which the access network transmits F-BPICH.
FLReservedInterlaces	This field shall determine what interlaces contain reserved bandwidth on the forward link (see Table 285).

Table 285—Interpretation of FLReservedInterlaces

Value	Interpretation: Reserved FL bandwidth on the following interlaces
0000	None
0001	0
0010	0, 1
0011	0, 1, 2
0100	0, 1, 2, 3
0101	0, 1, 2, 3, 4
0110	0, 1, 2, 3, 4, 5
0111	0, 1, 2, 3, 4, 5, 6
1000	0, 1, 2, 3, 4, 5, 6, 7
1001	0, 3
1010	0, 6
1011	0, 2, 4
1100	0, 2, 4, 6
1101	Reserved
1110	Reserved
1111	Reserved

NumFLReservedSubzones	This field determines the number of subzones that are reserved on each frame that contains reserved bandwidth.
Reserved	The access terminal shall ignore this field.

11.6.5.3 QuickChannelInfo Block

The QuickChannelInfo block shall be transmitted directly by the Superframe Preamble MAC Protocol over the SBCCH channel, and shall not pass through other Layers. The QuickChannelInfo block shall have the following format (see Table 286).

Table 286—QuickChannelInfo Block

Field	Length (bits)
CPICHoppingMode	1
NumEffectiveAntennas	3
FLSubzoneSize	1
ResourceChannelMuxMode	1
NumDRCHSubzones	5
BRCHSubzoneCyclingEnabled	1
UseDRCHforFLCS	1
NumCommonSegmentHopPortBlocks	4 or 5
NumLABSegments	2 or 3
MinSCCHResourceIndex	5
SinglePAForMultipleChannelband	1
FLNumSDMADimensions	2
ModulationSymbolsPerQPSKLAB	3
CommonPilotSpacing	1
EnableExpandedQPCH	1
Padding	Variable

CPICHoppingMode This field shall be set to 0 for deterministic, and 1 for random hopping.

NumEffectiveAntennas This field shall determine the effective number of antennas, and this parameter shall take the value $n + 1$. The access network shall set EffectiveNumAntennas field to less than four if ResourceChannelMuxMode = 0, or if ResourceChannelMuxMode = 1 and NumDRCHSubzones > 0.

FLSubzoneSize This field shall determine the number of subzones on the forward link. If $n = 0$, this parameter shall be set to 64 and if $n=1$, this parameter shall be set to 128.

ResourceChannelMuxMode This field shall be used by the Physical Layer to determine the hop pattern for the sector. This field shall be set to '0' to indicate MuxMode1 and to '1' to indicate MuxMode2.

NumDRCHSubzones This field shall take values between 0 and $N_{FFT}/64 - 1$.

BRCHSubzoneCyclingEnabled

This field determines if the BRCH subzones are cyclically shifted across interlaces in ResourceChannelMultiplexingMode 2.

UseDRCHforFLCS

This field shall be set to 1 if the hopping pattern is DRCH on the Forward Link Control Segment, and shall be set to 0 otherwise.

NumCommonSegmentHopPortBlocks

The length of this field shall be 4 bits if UseDRCHForFLCS is equal to '0', and 5 bits if UseDRCHForFLCS is equal to '1'. This parameter shall take the value 'n + 1'. For values of NFFT less than or equal to 1024, this field shall take values in the range 0–8.

NumLABSegments

The length of this field shall be 2 bits if UseDRCHForFLCS is equal to '0', and 3 bits if UseDRCHForFLCS is equal to '1'. This field shall indicate the number of LABSegments. This parameter shall take the value n if UseDRCHForFLCS = 0, and this parameter is not defined if UseDRCHForFLCS = 1.

MinSCCHResourceIndex

This field shall be in units of $\text{ceil}(18 \times (N_{\text{FFT}}/2048))$ resources, and spans from 0 to $N_{\text{FFT}} - 1$.

SinglePAForMultipleChannelBands

This field shall determine if the sector making a F-BPICH on one carrier blanks transmissions on all other carriers during that OFDM symbol.

FLNumSDMADimensions

This field shall determine the number of subtrees on the forward link. This parameter takes the value 'n + 1'.

ModulationSymbolsPerQPSKLAB

This field shall determine the number of modulation symbols for each block carried by the F-SCCH (see Table 287).

Table 287—Interpretation of ModulationSymbolsPerQPSKLAB

Value	Interpretation
000	35
001	50
010	80
011	110
100	180
Other values	Reserved

CommonPilotSpacing

This parameter shall take the value $8 \times (n + 1)$.

EnableExpandedQPCH

This field determines whether more than one packet is delivered to the Physical Layer by the Lower MAC Sublayer. This field shall be set to 1 if more than one packet is delivered to the Physical Layer by the MAC Layer for transmission on F-QPCH, and shall be set to 0 otherwise.

Padding The transmitter shall set the number of reserved bits to make the transmitted packet (including the CRC added by the Physical Layer) octet aligned. The access terminal shall ignore this field. The access network shall set this field to zero.

11.6.5.4 ExtendedChannelInfo

The ExtendedChannelInfo message provides the configuration for the Physical Layer and Lower MAC Sublayer operation. The message consists of several groups, and a group consists of fields, as defined in the following (see Table 288).

Table 288—ExtendedChannelInfo message

Field	Length (bits)
MessageID	8
ECISignature	24
ChannelInfoValidityIndex	4
SystemTime	37
SectorParametersSignature	16
ForwardChannelGroup	See 11.6.5.4.1
ReverseChannelGroup	See 11.6.5.4.2
PowerControlGroup	See 11.6.5.4.3
AccessParametersGroup	See 11.6.5.4.4
Padding	Variable

MessageID This field shall be set to 0x00.

ECISignature The access network shall change this field when it requires the access terminal to read the ExtendedChannelInfo message again. This field is used to scramble the F-QPCH channel. In case soft handoff is being used on F-QPCH, the access network should set the same ECISignature for all sectors in the soft handoff group. When the sector begins providing service, the access network should set this field to a random number. To prevent continuous conflict between the ECISignatures of sectors, when this field is changed, the access network should increment this field by a random number.

ChannelInfoValidityIndex This field determines the time until the parameters in the ExtendedChannelInfo message and the QuickChannelInfo block are valid.

SystemTime This field shall be set to the time at the sector this message refers to at the beginning of the superframe in which the ExtendedChannelInfo block started transmission.

SectorParametersSignature The access network shall set this field to the SectorParametersSignature of the next SectorParameters message to be transmitted by the access network.

ForwardChannelGroup This field is defined in 11.6.5.4.1.

ReverseChannelGroup This field is defined in 11.6.5.4.2.

PowerControlGroup This field is defined in 11.6.5.4.3.

AccessParametersGroup This field is defined in 11.6.5.4.4.

Padding The number of bits in this field is equal to the number needed to make the message length an integer number of octets. The access network shall set this field to zero. The access terminal shall ignore this field.

Channels	FTC

RLP	Best effort

Addressing	Broadcast	Unicast

AuthTag	Not included

11.6.5.4.1 ForwardChannelGroup

This group, together with the QuickChannelInfo block, determines the structure of the forward channels (see Table 289).

Table 289—ForwardChannel Group

Field	Length (bits)
CQIPilotTransmitPower	4
CommonPilotTransmitPower	4
FAckBandwidthFactor	2
MaxNumCDMAFAckIndices	3
FLDPICHCodeOffsetSubtreeIndex _j , $j = 1, 2, 3$	2
SFNCeIIIDSameAsPilotID	1
SFNCeIIID	9 ^a
CellNullIIDIncluded	1
CellNullIID	9 ^a
MaxNumSharedLABs	2
16QAMIncluded	1
Max16QAMSCCHBlocks	4 ^a
16QAMSCCHT2PRatio	2 ^a
PDCABResourceSharingEnabled	1
NumAckableLABs	3
EffectiveTransmitPower	6

^aThis field may be absent (see field description for conditions).

CQIPilotTransmitPower The field shall determine the power spectral density of the F-CQIPICH relative to the reference transmit power density defined by the Physical Layer. This parameter shall take the value $(-4 + n \times 0.5)$ dB.

CommonPilotTransmitPower

The field determines the power spectral density of the F-CPICH during the FL PHY frame. This parameter shall take the value $(-4 + n \times 0.5)$ dB.

FAckBandwidthFactor This parameter takes the value 2^n .

MaxNumCDMAFAckIndices

This field determines the number of FACK indices on a frame for RL CDMA traffic. This parameter takes the values [0, 8, 16, 32, 48, 64, 96, 128], depending on the value of the field.

FLDPICHCodeOffsetSubtreeIndex j

This field shall be set to the corresponding value for subtree j . The access network should set this field to zero for values of j greater than FLNumSDMADimensions.

SFNCelIDSameAsPilotID

This field determines if the SFNCelID is the same as the PilotID for this sector. The access network shall set SFNCelIDSameAsPilotID to '1' if EnablePreambleFrequencyReuse is equal to '1'.

SFNCelID

This field shall be included if SFNCelIDSameAsPilotID is equal to 0, and shall be omitted otherwise. If this field is omitted, the access terminal shall set the parameter SFNPN to the lower 9 bits of the PilotID.

CellNullIDIncluded

This field determines if F-CNCH is transmitted for this sector.

CellNullID

This field shall be included if CellNullIDIncluded is equal to 0, and shall be omitted otherwise.

MaxNumSharedLABs

This field shall determine the maximum number of shared LABs that are transmitted by this sector. This parameter takes the value $n + 2$.

16QAMIncluded

This field shall determine if the Max16QAMSCCHBlocks and 16QAMSCCHT2PRatio fields are included in the message.

Max16QAMSCCHBlocks

This field determines the maximum number of 16-QAM blocks that may be transmitted by the access network. If this 16QAMIncluded = 1, this parameter shall take the value $2 \times n$; otherwise this field shall be omitted and this parameter shall be set to 0.

16QAMSCCHT2PRatio

This field shall be included if 16QAMIncluded = 1, and shall be omitted otherwise. If included, this parameter shall take the values -10 dB, -7 dB, -4 dB, and 0 dB, depending on this field taking values '00', '01', '10', or '11' respectively.

PDCABResourceSharingEnabled

This field determines if resource sharing using PDCAMs is enabled.

NumAckableLABs

This field shall be set to the number of LABs on SCCH that the access terminal shall acknowledge.

EffectiveTransmitPower

This field shall be used by the access terminal to calculate ChanDiff as defined in the MAC Layer. This parameter takes the value $20 + n$ dBm. The access network

should set this field based on the transmit power of the sector and other quantities such as noise figure.

11.6.5.4.2 ReverseChannelGroup

This group, together with the QuickChannelInfo block, determines the structure of the reverse channels (see Table 290).

Table 290—ReverseChannel Group

Field	Length (bits)
AssignmentAckHARQTransmission	3
BFCHReportingMode	1
CDMAInterlacesBitmap	8
Bitcount(CDMAInterlacesBitmap) occurrences of the following field {	
NumCDMASubsegments _j	4
}	
Sum(NumCDMASubsegments _j) occurrences of the following field {	
RoT _j	3
}	
NumODCCHReports	5
RACKBandwidthFactor	2
RLNumSDMADimensions	2
RLNumSDMADimensions occurrences of the following field	
RLDPICHCodeOffsetSubtreeIndex _j	2
RLSubzoneSize	1
SilenceIntervalPeriod	4
SilenceIntervalDuration	3
NumSilenceIntervalSubzones	4

AssignmentAckHARQTransmission

The value '000' indicates that no ACK is sent in response to an assignment. The rules for interpreting other values of this field are provided in the Lower MAC Sublayer. The value '111' is reserved.

BFCHReportingMode

This field shall be set to 0 for non-SDMA and 1 for SDMA reporting modes.

CDMAInterlacesBitmap

The j^{th} bit of this field shall be set to 1 if interlace i contains a Reverse Link CDMA Segment. Here j is assumed to range from 0 through 7 and an interlace i is the set of PHY frames that satisfy $\text{PHY frame Index mod } 8 = i$.

NumCDMASubsegmentsInterlace_j

This field shall be included if the j^{th} bit of the field CDMAInterlacesBitmap is

equal to 1, and shall be omitted otherwise. This field determines the number of subsegments in interlace j . This parameter takes the value ' $n + 1$ '.

RoT j This field shall be the RoT in units of dB on SubSegment k , where k indexes over the available $\text{Sum}(\text{NumCDMASubsegments}_j)$ CDMA Subsegments. This parameter shall take the value 0, 1, 2, 3, 4, 6, 8, 10 dB, depending on whether the field is set to 0, 1, 2, 3, 4, 5, 6, or 7 respectively.

NumODCCHReports This field shall be specified in units of 16.

RACKBandwidthFactor This parameter shall be set to 2^n , where n is the value of the two bit field.

RLNumSDMADimensions This field shall determine the number of spatial dimensions on the reverse link. This parameter shall take the value $n + 1$.

RLDPICHCodeOffsetSubtreeIndex j The j^{th} occurrence of this field shall be set to the code offset for tree j , where $j = 1, \dots, \text{RLNumSDMADimensions} + 1$. **RLDPICHCodeOffsetSubtreeIndex0** is always equal to 0.

RLSubzoneSize This field shall determine the size of subzones on the reverse link. If $n=0$, this parameter shall take the value 64 and if $n = 1$, this parameter shall take the value 128.

SilenceIntervalPeriod This field shall determine the period in units of superframes when the silence interval repeats. The **SilenceInterval** shall take values 2^n superframes, where n is the value of this four bit field.

SilenceIntervalDuration This field shall determine the duration silence interval in units of PHY frames. The duration of the silence interval shall be 0 if this field takes the value 0. Otherwise, this parameter takes the value $2^{(n-1)}$.

NumSilenceIntervalSubzone This field specifies the set of subzones that are blanked during the silence interval.

11.6.5.4.3 PowerControlGroup

This group contains parameters used for access terminal power control (see Table 291).

Table 291—PowerControlGroup

Field	Length (bits)
ACKInterferenceOffset	2
MACIDRange	3
FLPCReportInterval	2
FLPQIReportInterval	2
FLIOTReportInterval	2
IoTEnabled	1

Table 291—PowerControlGroup (continued)

Field	Length (bits)
FastOSIEnabled	1
RABEnabled	1
OSIResponseMode	1
SlowInterferenceOffset	4
CtrlAccessOffset	2

ACKInterferenceOffset This parameter shall take the value $2 \times n$ in units of dB.

MACIDRange This field shall be set to indicate the range of assigned MACID values in the sector. For example, a MACIDRange of 63 indicates that the sector has not assigned MACID values 64 and above. The field shall be interpreted as follows in Table 292.

Table 292—Interpretation of MACIDRange

Value	Interpretation
000	31
001	63
010	127
011	255
100	511
101	1023
110	2047
111	Reserved

FLPCReportInterval This field shall determine the periodicity at which power control commands are sent to the access terminal. This parameter shall take the value $2^{(n+2)}$, where n is the value of the two bit field.

FLPQIReportInterval This field shall determine the periodicity at which PQI reports commands are sent by this sector. This parameter shall take the value 16×2^n , where n is the value of the three bit field.

FLIOTReportInterval This field shall determine the periodicity at which IoT values are sent to the access terminal. This parameter shall take the value 1, 8, 16, or 32, depending on the two bit field taking values 0, 1, 2, or 3.

IoTEnabled This field shall determine if this sector transmits IoT.

FastOSIEnabled This field shall determine if the access terminal is required to read OSI from this sector. This field shall be set to '1' if the F-SCCH transmitted by this sector contains a Fast OSI Segment. This field shall be set to '0' if the F-SCCH transmitted by this sector does not contain a Fast OSI Segment.

RABEnabled	This field shall be set to '1' if this sector transmits RAB, and shall be set to '0' otherwise.
OSIResponseMode	This field shall be set to 0 for stochastic, and 1 for deterministic response to OSI modes.
SlowInterferenceOffset	This field shall be set in units of dB.
CtrlAccessOffset	This field determines the initial gain of the R-CQICH over the R-ACH. The value of this parameter shall be $-11 + 2 \times n$ dB, where n is the value of this field.

11.6.5.4.4 AccessParameters Group

This group contains parameters used for access as shown in Table 293.

Table 293—AccessParameters Group

Field	Length (bits)
AccessCycleDuration	1
MaxProbesPerSequence	3
ProbeRampUpStepSize	2
PilotThreshold1	3
PilotThreshold2	3
OpenLoopAdjust	8
InitialAccessSubsegment	4
SpecialAccessSubsegment	4
AccessRetryPersistence0	3
$N_{ACMPClass} - 1$ values of the following field	
AccessRetryPersistenceSameAsPreviousj	1
AccessRetryPersistencej	0 or 3

AccessCycleDuration	This field shall determine the duration of the access cycle in units of Access Opportunities (as defined by the Physical Layer). The AccessCycleDuration parameter shall be set according to the value of the field as follows in Table 294.

Table 294—Interpretation of AccessCycleDuration

Value	Interpretation
0	1
1	2

MaxProbesPerSequence	This field shall determine the maximum number of probe sequences that can be part of one access sequence. This parameter shall take the value $n + 2$.

- ProbeRampUpStepSize** This field shall determine the power ramp up used for probes within a probe sequence. This parameter shall take the value $2 \times (1 + n)$ dB.
- PilotThreshold1** This field shall determine PilotThreshold1 used by the Access Channel MAC Protocol. This parameter shall take the value $-10 + 2n$ dB.
- PilotThreshold2** This field shall determine PilotThreshold2 used by the Access Channel MAC Protocol. This parameter shall take the value $-2n$ dB.
- OpenLoopAdjust** This field shall determine the nominal power to be used by access terminal in the open-loop power estimate. The value of this field shall be $70 + n$ dB. This group provides parameters required to use the Access Channel (R-ACH)
- InitialAccessSubsegment** This field shall be set to the CDMA Subsegment that the access terminal may use for initial access.
- SpecialAccessSubsegment** This field shall be set to the CDMA Subsegment that the access terminal may use for sending access probes after it has acquired a MAC ID.
- AccessRetryPersistence0** This field shall determine AccessRetryPersistence0 for determining access sequence backoff. For n less than 7, if this field is set to n , the access terminal shall set the parameter AccessRertyPersistence0 to $2^{-n/2}$. For $n=7$, the access terminal shall set AccessRetryPersistence j to 0.
- AccessRetryPersistanceSameAsPrevious j** The access network shall set this field to 0 if the AccessRetryPersistence j is same as AccessRetryPersistence($j - 1$), and to 1 otherwise.
- AccessRetryPersistance** This field shall be omitted if AccessRetryPersistanceSameAsPrevious is equal to 0, and shall be included otherwise. For n less than 7, if this field is set to n , the access terminal shall set AccessRetryPersistence j to $2^{-n/2}$. For $n = 7$, the access terminal shall set AccessRetryPersistence j to 0.

11.6.5.5 SectorParameters

The SectorParameters message is used to convey sector specific information to the access terminals. This message may also include information about neighboring sectors (see Table 295).

Table 295—SectorParameters message

Field	Length (bits)
MessageID	8
MobileCountryCode	12
MobileNetworkCode	12
SectorID	128
SectorParametersSignature	16
PilotIDIncluded	1
PilotID	10 ^a
ChannelBand	Variable

Table 295—SectorParameters message (continued)

Field	Length (bits)
Latitude	22
Longitude	23
LeapSeconds	8
LocalTimeOffset	11
PrimaryRegistrationZoneCode	8
NumSecondaryRegistrationZoneCodes	4
NumSecondaryRegistrationZoneCodes instances of the following field{	
SecondaryRegistrationZoneCode	8
}	
FastRepageSupported	1
PilotGroupIDIncluded	1
PilotGroupID	0 or 3
CellGroupIDIncluded	1
CellGroupID	3 ^a
NumSupportedIPSI	3
NumSupportedIPSI occurrences of the following record{	
SupportedIPSI	4
}	
NumSupportedBCMCSIPSI	2
NumSupportedBCMCSIPSI instances of the following field{	
BCMCSIPSI	4
}	
ChannelListCount	5
ChannelListCount occurrences of the following record{	
ChannelBand	Variable
ShortChannelIDIncluded	1
ShortChannelID	2
AccessLoadControlIncluded	1
AccessLoadControl	3 ^a
AccessHashingMaskLength	4
AccessHashingChannelMaskSameAsPrevious	1
AccessHashingChannelMask	0 or AccessHashingMaskLength + 1
}	

Table 295—SectorParameters message (continued)

Field	Length (bits)
SynchronousCPListCount	3
SynchronousCPListCount occurrences of the following field	
SynchronousCP	2
AsynchronousCPListCount	3
AsynchronousCPListCount occurrences of the following field	
AsynchronousCPList	2
NeighborCount	5
NeighborCount occurrences of the following record{	
NeighborPilotID	10
EffectiveTransmitPowerIncluded	1
EffectiveTransmitPower	6
NeighborChannelSameAsPrevious	1
NeighborChannel	Variable
NeighborChannelShortIDIncluded	1
NeighborChannelShortID	2
SameANAsPrimarySector	1
PilotGroupIDIncluded	1
PilotGroupID	3 ^a
SynchronousGroupID	3 ^a
CellGroupIDIncluded	1
CellGroupID	3 ^a
}	
NumNeighborTechnologies	4
NumNeighborTechnologies occurrences of the following fields {	
TechnologyType	8
NeighborTechnologyListLength	8
NeighborTechnologyList	NeighborTechnologyListLength × 8
Padding	Variable

^aThis field may be absent (see field description for conditions).

MessageID The access network shall set this field to 0x01.

MobileCountryCode The access network shall set this field to the three-digit BCD (binary coded decimal) encoded representation of the Mobile Country Code associated with this sector (the primary sector). The country code is defined in ITU-T

Recommendation E.212, The International Identification Plan for Mobile Terminals and Mobile Users, 2004.

MobileNetworkCode	This field shall be set three-digit BCD (binary coded decimal) encoded representation of the Mobile Network Code that has been assigned to the operator.
SectorID	Sector Address Identifier. The access network shall set the value of the SectorID according to the rules specified in 16.2.2. The access terminal shall not assume anything about the format of the SectorID. The SectorID shall uniquely identify a sector.
SectorParametersSignature	SectorParameters message signature. The access network should change this field if the contents of the SectorParameters message changes, except for changes to the following fields: — AccessLoadControl for other channels.
PilotIDIncluded	This field shall determine if the following PilotID field is included in this message.
Pilot ID	This field shall be included if PilotIDIncluded is '1', and shall be omitted otherwise. The access network shall set this field to the PilotID of the sector that the information in this message corresponds to.
ChannelBand	ChannelBand record specification for each channel (see 15.2.1).
Latitude	The latitude of the sector. This field shall be included if LatitudeLongitudeIncluded is equal to 1, and shall be omitted otherwise. The access network shall set this field to this sector's latitude in units of 0.25 second, expressed as a two's complement signed number with positive numbers signifying North latitudes. The access network shall set this field to a value in the range $-1\ 296\ 000$ to $1\ 296\ 000$ inclusive (corresponding to a range of -90° to $+90^\circ$).
Longitude	The longitude of the sector. This field shall be included if LatitudeLongitudeIncluded is equal to 1, and shall be omitted otherwise. The access network shall set this field to this sector's longitude in units of 0.25 second, expressed as a two's complement signed number with positive numbers signifying East longitude. The access network shall set this field to a value in the range $-2\ 592\ 000$ to $2\ 592\ 000$ inclusive (corresponding to a range of -180° to $+180^\circ$).
LeapSeconds	The number of leap seconds that have occurred since the start of system time.
LocalTimeOffset	The access network shall set this field to the offset of the local time from System Time. This value shall be encoded in units of minutes, expressed as a two's complement signed number.
PrimaryRegistrationZoneCode	The PrimaryRegistrationZoneCode for this sector.

NumSecondaryRegistrationZoneCodes	The number of SecondaryRegistrationZoneCode values advertised by this sector.
SecondaryRegistrationZoneCode	One of the SecondaryRegistrationZoneCode for this sector.
FastRepageSupported	The access network shall set this field to '1' if this sector transmits fast repages. The access network shall set this field to '0' otherwise.
PilotGroupIDIncluded	The access network shall set this field to '1' if PilotGroupID field for this sector is included in this message. Otherwise, the access network shall set this field to '0'.
PilotGroupID	The access network shall omit this field if PilotGroupIDIncluded is set to '0'. If PilotGroupIDIncluded is set to '1', the access network shall set this field to the PilotGroupID associated with the neighbor pilot.
CellGroupIDIncluded	The access network shall set this field to '1' if CellGroupID field for this sector is included in this message. Otherwise, the access network shall set this field to '0'.
CellGroupID	The access network shall omit this field if CellGroupIDIncluded is set to '0'. If CellGroupIDIncluded is set to '1', the access network shall set this field to the CellGroupID associated with the neighbor pilot. The CellGroupID fields carry information about which sectors included in this message belong to the same cell, and the fields values by themselves have no meaning across different occurrences of this message. The access network shall assign the same CellGroupID to two sectors included in this message if the two sectors belong to the same cell. The access network shall assign different CellGroupIDs to two sectors included in this message if the two sectors belong to different cells. The CellGroupID field shall be omitted for sectors whose cell membership is not known to the sector transmitting this message.
NumSupportedIPSI	The access network shall set this field to the number of IPSIs supported by this sector.
SupportedIPSI	The access network shall set this field to an IPSI supported by this sector.
NumSupportedBCMCSIPSI	The access network shall set this field to the number of BCMCS IPSIs supported by this sector.
BCMCSIPSI	The access network shall set this field to a BCMCS IPSI supported by this sector.
ChannelListCount	The access network shall set this field to the number of neighbor channels available to the access terminal on this sector.
ChannelBand	ChannelBand record specification for each channel (see 15.2.1). If the optional parameters of this ChannelBand record are different from the parameters for the sector transmitting this message, the access network shall include the optional parameters in the ChannelBand record.
ShortChannelIDIncluded	This field shall determine if the ShortChannelID field is included for this ChannelBand.

ShortChannelID	This field shall be included if ShortChannelIDIncluded equals 1, and shall be omitted otherwise. This field identifies the two bit index that identifies this channel in beacon transmissions. The access network should keep the mapping between ShortChannelID and ChannelBand the same over a wide geographic area.
AccessLoadControlIncluded	This field determines if the AccessLoadControl for this channel is included. If this field is set to '1', the AccessLoadControl field is included. If this field is set to '0', the AccessLoadControl field is not included.
AccessLoadControl	If the AccessLoadControlIncluded field is set to '0', the access network shall omit this field. Otherwise, this field is set to the access loading control threshold, which is associated with the access classes for this channel.
AccessHashingMaskLength	The access network shall set this field one less than the number of bits in the AccessHashingChannelMask field(s).
AccessHashingChannelMaskSameAsPrevious	If the AccessHashingChannelMask for the i^{th} ChannelBand in this message is the same as the AccessHashingChannelMask for the $(i-1)^{\text{th}}$ ChannelBand, the access network shall set this field to one. Otherwise, the access network shall set this field to '0'.
AccessHashingChannelMask	The access network shall set this field in the i^{th} occurrence of this record as follows: If the AccessHashingChannelMaskSameAsPrevious field in this record is set to '1', the access network shall omit this field. Otherwise, the access network shall set this field to the $(\text{AccessHashingMaskLength} + 1)$ bit access hashing class of the i^{th} combined channel list entry in this message. If this field is not included, the access terminal shall assume that the value of this field is the same as the value for this field in the previous occurrence of this record.
SynchronousCPListCount	This field shall be set to the number of SynchronousCPj fields included in this message. The sector transmitting this message should include the cyclic prefix of neighboring GlobalSynchronous sectors in the SynchronousCPList.
SynchronousCP	This field shall be set to one less than the Cyclic Prefix of an IsSynchronous neighbor sector. The Cyclic Prefix is defined by the Physical Layer.
AsynchronousCPListCount	This field shall be set to the number of AsynchronousCPj fields included in this message. The sector transmitting this message should include the cyclic prefix of neighboring non-IsSynchronous sectors in the SynchronousCPList.
AsynchronousCP	This field shall be set to one less than the Cyclic Prefix of a non-IsSynchronous neighbor sector. The Cyclic Prefix is defined by the Physical Layer.
NeighborCount	The access network shall set this field to the number of records specifying neighboring sectors information included in this message for this channel.

NeighborPilotID	The access network shall set this field to the PilotID of a neighboring sector that the access terminal should add to its Neighbor Set.
EffectiveTransmitPowerIncluded	This field determines if the EffectiveTransmitPower for this neighbor is included.
EffectiveTransmitPower	This field shall be included if EffectiveTransmitPowerIncluded is equal to 1, and shall be omitted otherwise. This field shall be used by the access terminal to calculate ChanDiff and other related quantities at the MAC Layer. This parameter shall take the value $20 + n$ dBm. The access network should set this field based on the transmit power of the sector and other quantities such as noise figure.
NeighborChannelSameAsPrevious	This field determines if the NeighborChannel, field is included.
NeighborChannel	This field shall be included if NeighborChannelSameAsPrevious is equal to 0, and shall be omitted otherwise. ChannelBand record specification for each channel. If included, this field shall be set to the channel record for this pilot.
NeighborChannelShortIDIncluded	This field shall be included if ChannelBandIncluded is equal to 1, and shall be omitted otherwise. This field shall determine if the NeighborChannelShortID field is included for this ChannelBand.
NeighborChannelShortID	This field shall be included if ChannelBandIncluded and NeighborChannelShortIDIncluded are both equal to 1, and shall be omitted otherwise. This field identifies the two bit index that identifies this Channel in beacon transmissions.
SameANAsPrimarySector	This field shall be set to 1 if this pilot of the neighbor list belongs to the same AN as the primary sector transmitting this message.
PilotGroupIDIncluded	The access network shall set this field to '1' if PilotGroupID field for this sector is included in this message. Otherwise, the access network shall set this field to '0'.
PilotGroupID	The access network shall omit this field if PilotGroupIDIncluded is set to '0'. If PilotGroupIDIncluded is set to '1', the access network shall set this field to the PilotGroupID associated with the neighbor pilot.
CellGroupIDIncluded	The access network shall set this field to '1' if CellGroupID field for this sector is included in this message. Otherwise, the access network shall set this field to '0'.
CellGroupID	The access network shall omit this field if CellGroupIDIncluded is set to '0'. If CellGroupIDIncluded is set to '1', the access network shall set this field to the CellGroupID associated with the neighbor pilot. The CellGroupID fields carry information about which sectors included in this message belong to the same cell, and the fields values by themselves have no meaning across different occurrences of this message. The access network shall assign the same CellGroupID to two sectors included in this message if the two sectors belong to the same cell. The access network shall assign different CellGroupID fields to two sectors included in this message if the two sectors belong to different cells.

The CellGroupID field shall be omitted for sectors whose cell membership is not known to the sector transmitting this message.

NumNeighborTechnologies

This field shall be set to the number of neighbor technologies included in the message. Neighbor technologies are included in this message to assist the access terminal in inter-technology handoff.

TechnologyType

This field shall be set to the type of technology, and shall be interpreted as specified in 15.2.2.

NeighborTechnologyRecordLength

This field shall be set the length, in bytes, of the neighbor technology record for the neighbor technology.

NeighborTechnologyRecord

This field shall be set to the neighbor technology record for the neighbor technology. The interpretation of this field is specified in 15.2.2.

Padding

The number of bits in this field is equal to the number needed to make the message length an integer number of octets. The access network shall set this field to zero. The access terminal shall ignore this field.

Channels	FTC	
Addressing	Broadcast	Unicast

RLP	Best effort
AuthTag	Not required

11.6.5.6 EncapsulatedQuickChannelInfo

This message may be used to deliver the QuickChannelInfo block over the Forward Traffic Channel MAC. This message may also be used to deliver the QuickChannelInfo block of one sector over the Forward Traffic Channel of another sector (see Table 296).

Table 296—EncapsulatedQuickChannelInfo message

Field	Length (bits)
MessageID	8
EffectiveSuperframeNumber	37
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
QuickChannelInfoLength	8
QuickChannelInfoBlock	QuickChannelInfoLength
Padding	Variable

MessageID

This field shall be set to 0x02.

EffectiveSuperframeNumber

This field shall be set to the superframe number at the sector this message refers to, when that sector transmitted the QuickChannelBlock included in this message over the Superframe Preamble MAC.

PilotID

This field shall be set to the PilotID of the sector to which this message refers. The sector this message refers to may be different from the sector transmitting this message.

ChannelBandIncluded

This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message.

ChannelBand

This field shall be included if and only if ChannelBand Included is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers.

QuickChannelInfoLength This field shall be set to the length, in bits, of the following QuickChannelInfoBlock.

QuickChannelInfoBlock This field shall be set to the QuickChannelBlock.

Padding

The number of bits in this field is equal to the number needed to make the message length an integer number of octets. The access network shall set this field to zero. The access terminal shall ignore this field.

Channels	FTC	
Addressing	Broadcast	Unicast

RLP	Best effort
AuthTag	Required if key available

11.6.5.7 EncapsulatedExtendedChannelInfo

This message may be used to deliver the ExtendedChannelInfo message over the Forward Traffic Channel MAC at times that do not correspond to normal delivery of ExtendedChannelInfo. This message may be used to improve the reliability of delivering new overhead parameters when the overhead parameters are changing. This message may also be used to deliver the QuickChannelInfo block of one sector over the Forward Traffic Channel of another sector (see Table 297).

Table 297—EncapsulatedExtendedChannelInfo message

Field	Length (bits)
MessageID	8
EffectiveSuperframeNumber	37
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
ECISignatureIncluded	1
Reserved	2
ECISignature	24 ^a

Table 297—EncapsulatedExtendedChannelInfo message (continued)

Field	Length (bits)
ExtendedChannelInfoLength	8
ExtendedChannelInfoMessage	$8 \times \text{ExtendedChannelInfo Length}$

^aThis field may be absent (see field description for conditions).

MessageID	This field shall be set to 0x03.
EffectiveSuperframeNumber	This field shall be set to the superframe number when the source sector transmits the parameters included in this message in the form of an ExtendedChannelInfo message.
PilotID	This field shall be set to the PilotID of the sector to which this message refers. The sector this message refers to may be different from the sector transmitting this message.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.
ECISignatureIncluded	This field shall determine if the ECISignature is included in the message. The access network may set this field to '1' in case this message is being transmitted to access terminals that have received an earlier copy of the ExtendedChannelInfo message, and the sector configuration has not changed after the expiry of the validity timer.
Reserved	The access terminal shall ignore this field.
ECISignature	This field shall be included if ECISignatureIncluded is equal to '1', and is set to the ECISignature that will be effective after superframe number EffectiveSuperframeNumber. This field shall be omitted if ECISignatureIncluded is equal to '0'.
ExtendedChannelInfoLength	This field shall be set to the length, in bytes, of the following ExtendedChannelInfoMessage. This field shall be set to 0 if ECISignatureIncluded is equal to '1'.
ExtendedChannelInfoMessage	This field shall be included if and only if ECISignatureIncluded is equal to '0'. When present, this field shall be set to the ExtendedChannelInfo message.

Channels	FTC
Addressing	Broadcast Unicast

RLP	Best effort
AuthTag	Required if key available

11.6.5.8 EncapsulatedSystemInfo

This message may be used to deliver the SystemInfo block over the Forward Traffic Channel MAC. This message may also be used to deliver the SystemInfo block of one sector over the Forward Traffic Channel of another sector (see Table 298).

Table 298—EncapsulatedSystemInfo message

Field	Length (bits)
MessageID	8
EffectiveSuperframeNumber	37
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
SystemInfoLength	8
SystemInfoBlock	SystemInfoLength
Padding	Variable

MessageID	This field shall be set to 0x04.
EffectiveSuperframeNumber	This field shall be set to the superframe number at the sector this message refers to, when that sector transmitted the QuickChannelBlock included in this message over the Superframe Preamble MAC.
PilotID	This field shall be set to the PilotID of the sector to which this message refers. The sector this message refers to may be different from the sector transmitting this message.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.
SystemInfoLength	This field shall be set to the length, in bits, of the following SystemInfoBlock.
SystemInfoBlock	This field shall be set to the QuickChannelBlock.
Padding	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. The access network shall set this field to zero. The access terminal shall ignore this field.

Channels	FTC		RLP	Best effort
Addressing	Broadcast	Unicast	AuthTag	Required if key available

11.6.6 Interface to other protocols

11.6.6.1 Commands

This protocol does not issue any commands.

11.6.6.2 11.6.6.2 Indications

This protocol registers to receive the following indications:

- *ActiveSetManagement.IdleHO*
- *ConnectedState.ConnectionClosed*

11.6.7 Configuration attributes

This protocol does not define any attributes.

11.6.8 Protocol numeric constants

Protocol numeric constants are listed in Table 299.

Table 299—Protocol numeric constants

Constant	Meaning	Value
N _{OMPT} Type	Type field for this protocol	See 16.1
N _{OMP} Basic	Subtype field for this protocol	0x0000
T _{OMPCIS} Supervision	ChannelInfo supervision timer	80 superframes
T _{OMPSPS} Supervision	SectorParameters supervision timer	320 superframes
N _{OMPECI}	The time between two consecutive ExtendedChannelInfo message transmissions	8 superframes
N _{OMP} SectorParameters	The time between two consecutive SectorParameters message transmissions	64 superframes

11.6.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

The parameter records for this protocol consist of the configuration attributes of this protocol.

11.7 Basic Active Set Management Protocol

11.7.1 Overview

The Basic Active Set Management Protocol provides the procedures and messages used by the access terminal and the access network to maintain the radio link as the access terminal moves between the coverage areas of different sectors.

This protocol can be in one of three states:

- *Inactive State*: In this state, the protocol waits for an Activate command.
- *Idle State*: This state corresponds to the Air Link Management Protocol Idle State. In this state, the access terminal autonomously maintains the Active Set.
- *Connected State*: This state corresponds to the Air Link Management Protocol Suspended or Connected States. In this state, the access network dictates the access terminal's Active Set based on PilotReports sent by the access terminal. PilotReport messages from the access terminal to the access network are based on changing radio link conditions.

Transitions between states are driven by commands received from other Connection Control Layer protocols and the transmission and reception of the MACResourceAssignment message.

The protocol states, messages and commands causing the transition between the states are shown in Figure 156.

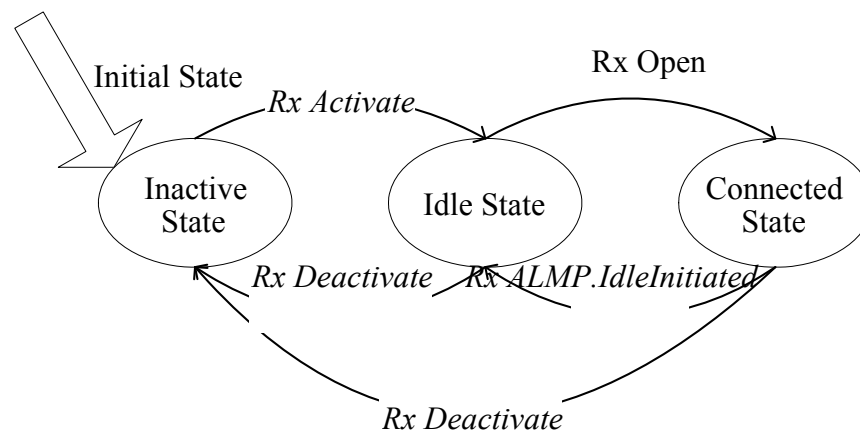


Figure 156—Basic Active Set Management Protocol State Diagram

This protocol uses parameters that are provided as public data by the Overhead Messages Protocol, configured attributes, or protocol constants.

11.7.2 Primitives

11.7.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

- *Open*

11.7.2.2 Return indications

This protocol returns the following indications:

- *ConnectionLost* (access network only)
- *IdleHandoff*
- *MACResourceUpdated*
- *ConnectionOpened*

11.7.2.3 Local common data

This protocol defines the following local common data:

- Combined Active Set
- Candidate Set
- Remaining Set

11.7.2.4 Public data

This protocol defines the following public data:

- Active Set
- Current MACResourceAssignment message
- Current SupplementalConfigAssignment, CDMA Traffic Assignment, and OFDM Control Assignment messages
- A list of measured PilotStrengths and RxPowers indexed by PilotID
- AccessBasedHandoffRequired (one entry for each pilot in the Active Set)
- Subtype for this protocol

11.7.3 Procedures and messages for the InConfiguration instance of the protocol

11.7.3.1 Protocol initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

11.7.3.2 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

11.7.3.3 Message formats

The InConfiguration instance of this protocol define any messages.

11.7.4 Procedures for the InUse instance

11.7.4.1 Initialization and commit procedures

11.7.4.1.1 Initialization procedures

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The protocol at the access terminal shall enter the Inactive State.
- The protocol at the access network shall enter the Inactive State.

11.7.4.1.2 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed to execute the Hard Commit procedures.

11.7.4.1.3 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed to execute the Soft Commit procedures.

11.7.4.2 Command processing

11.7.4.2.1 Activate

If the protocol at the access network receives an *Activate* command in the Inactive State, the protocol shall transition to the Idle State.

If the protocol at the access terminal receives an *Activate* command in the Inactive State, the protocol shall transition to the Idle State.

If this command is received in any other state, it shall be ignored.

11.7.4.2.2 Deactivate

If the protocol receives a *Deactivate* command in the Inactive State, it shall be ignored.

If the protocol receives a *Deactivate* command in any other state, the access terminal and the access network shall transition to the Inactive State.

11.7.4.2.3 Open

If the protocol receives an *Open* command in the Idle State,

- The access terminal shall perform the following
 - Transition to the Connected State.
 - If this protocol at no other InUse instance is in the Connected State, set PilotReportMessageSequence to zero.
- The access network shall transition to the Connected State.

If *Open* is received in any other state, it shall be ignored.

11.7.4.3 Pilots and pilot sets

The access terminal estimates the strength of the Forward Channel transmitted by each sector in its neighborhood. This estimate is based on measuring the strength of the Forward Pilot Channel (specified by the pilot's PilotID and the pilot's ChannelBand), henceforth referred to as the pilot.

When this protocol is in the Connected State, the access terminal uses pilot strengths to decide when to generate PilotReport messages.

When this protocol is in the Idle State, the access terminal uses pilot strengths to decide which sector's Control Channel it monitors.

The following pilot sets are defined to support the Active Set Management process:³⁷

- *Active Set*: The set of pilots associated with the sectors currently serving the access terminal. The Active Set is maintained separately by each InUse instance, and only contains sectors from the access network corresponding to the InUse instance. When a connection is open, a sector is considered to be serving an access terminal when there is a MAC ID assigned to the access terminal in that sector. When a connection is not open, a sector is considered to be serving the access terminal when the access terminal is monitoring that sector's control channel for page reception and registration. Parameters for Active Set member sectors in connected state are sent as part of the MACResourceAssignment message, and placed in the public data of the Active Set Management Protocol.
- *Candidate Set*: The pilots that are not in the Active Set, but are received by the access terminal with sufficient strength to indicate that the sectors transmitting them are good candidates for inclusion in the Active Set. The access terminal maintains one Candidate Set that is common to all InUse instances.
- *Remaining Set*: The set of all possible pilots on the current channel assignment, excluding the pilots that are in any of the two previous sets. The access terminal maintains one Remaining Set that is common to all InUse instances.
- *Combined Active Set*: The access terminal also maintains a Combined Active Set that is the union of the Active Sets of all InUse instances. This Combined Active Set is provided to the MAC Layer, and the MAC Layer may refer to the Combined Active Set simply as the Active Set. The access terminal shall support a maximum Combined Active Set size of MaxSupportedActiveSetSize pilots.

At any given instant a pilot in a given ChannelBand is a member of exactly one set. The access terminal maintains all three sets. The access network maintains only the Active Set.

The access terminal complies with the following rules when searching for pilots, estimating the strength of a given pilot, and moving pilots between sets.

11.7.4.3.1 Pilot search and strength measurement

The access terminal shall continually search for pilots in the Connected State and whenever it is monitoring any channel in the Idle State. The access terminal shall search for pilots in all pilot sets. The exact algorithm and search priority used by the access terminal for pilot search is outside the scope of the specification.

The access terminal shall measure the strength of every pilot it searches. The access terminal shall base its pilot strength measurement on the PilotStrength specified in the Physical Layer. The access terminal shall place the measured PilotStrength in its public data.

³⁷In this context, a pilot identifies a sector.

11.7.4.3.2 Pilot drop timer maintenance

For each pilot, the access terminal shall maintain a pilot drop timer.

- The access terminal shall start a pilot drop timer for each pilot in the Candidate Set or the Active Set whenever the strength becomes less than the value specified by PilotDrop. The access terminal shall consider the timer to be expired after the time specified by PilotDropTimer.
- The access terminal shall reset and disable the timer whenever the strength of the pilot becomes greater than the value specified by PilotDrop.

Subclauses 11.7.4.3.4 and 11.7.4.7.3 specify the actions the access terminal takes when the pilot drop timer expires.

11.7.4.3.3 Active Set management

The access terminal shall support a maximum Active Set size of $N_{\text{ASMPActive}}$ pilots.

Rules for maintaining the Active Set are specific to each protocol state (see 11.7.4.6.1 and 11.7.4.7.1).

11.7.4.3.4 Candidate Set management

The access terminal shall support a maximum Candidate Set size of $N_{\text{ASMPCandidate}}$ pilots. The Candidate Set shall not include multiple pilots in the same Pilot Group. When adding a pilot to the Candidate Set from a certain Pilot Group, the access terminal may choose to add any pilot that belongs to the same Pilot Group. Also, the access terminal may replace at any time a pilot in the Candidate Set with another pilot that belongs to the same Pilot Group.

The access terminal shall add a pilot to the Candidate Set if one of the following conditions is met:

- The pilot does not belong to the same Pilot Group as one of the pilots that is already in the Active Set or Candidate Set and the strength of the pilot exceeds the value specified by PilotAdd.
- The Pilot is deleted from the Active Set but its pilot drop timer has not expired.

The access terminal shall delete a pilot from the Candidate Set if one of the following conditions is met:

- A pilot in the same Pilot Group is added to the Combined Active Set.
- Pilot's drop timer has expired.
- Pilot is added to the Candidate Set; and, as a consequence, the size of the Candidate Set exceeds $N_{\text{ASMPCandidate}}$. In this case, the access terminal shall delete the weakest pilot in the set. Pilot A is considered weaker than pilot B:
 - If pilot A has an active drop timer but pilot B does not.
 - If both pilots have an active drop timer and pilot A's drop timer is closer to expiration than pilot B's.
 - If neither of the pilots has an active drop timer and pilot A's strength is less than pilot B's.

11.7.4.3.5 Remaining Set management

The access terminal shall initialize the Remaining Set to contain all of the pilots that are not already members of any other set. In Connected State, the remaining set shall consist of pilots on at least all channels assigned in the MACResourceAssignment message.

The access terminal shall delete the pilot from the Remaining Set if it adds a pilot belonging to the same Pilot Group to another set.

The access terminal shall add a pilot to the Remaining Set if it deletes the pilot from any other set and if a pilot belonging to the same Pilot Group is not in the Active Set or Candidate Set.

11.7.4.3.6 Pilot Groups

Two pilots are defined to belong to the same Pilot Group if both the PilotID and the PilotGroupID associated with the two pilots are the same, and the two pilots belong to the same AN. The PilotGroupID for a pilot is determined as follows:

- The PilotGroupID associated with a pilot in the Active Set is specified in the MACResourceAssignment message.
- If public data of the Overhead Messages protocol specifies the PilotGroupID for a pilot that is not included in the most recently received MACResourceAssignment message, then that value is the PilotGroupID for that pilot.
- If public data of the Overhead Messages protocol does not specify the PilotGroupID for a pilot and there is no PilotGroupID specified for the pilot in the most recently received MACResourceAssignment message, then the value of the PilotGroupID for that pilot is assumed to be zero.

The access terminal shall associate a PilotGroupID with each pilot in the Remaining Set that is different from any other PilotGroupID associated with any other pilot.

11.7.4.3.7 Scheduler Groups

Each pilot in the Active Set belongs to a Scheduler Group.³⁸ Two pilots are defined to belong to the same Scheduler Group if the following condition is satisfied:

- The SchedulerGroupIDIncluded field of the MACResourceAssignment message that assigned the current Active Set is set to '1' and the SchedulerGroupID associated with the two pilots are the same.

If the MACResourceAssignment message does not specify the SchedulerGroupID for a pilot in the Active Set, then the SchedulerGroupID associated with that pilot is assumed to be a number that is different from all other SchedulerGroupID specified in the message.

11.7.4.4 Message Sequence Numbers

The access network shall validate all received PilotReport messages as specified in 11.7.4.4.1.

The access terminal shall validate all received MACResourceAssignment messages as specified in 11.7.4.4.2.

The PilotReport message and the MACResourceAssignment message carry a MessageSequence field that serves to flag duplicate or stale messages.

The MessageSequence field of the PilotReport message is independent of the MessageSequence field of the MACResourceAssignment message.

11.7.4.4.1 PilotReport Message Validation

When the access terminal sends a PilotReport message, it shall set the MessageSequence field of the message to one more than (modulo 2^S , $S = 8$) the MessageSequence of the previous PilotReport message.

³⁸For example, pilots in the same Scheduler Group share the same QNSequence space as specified by the Radio Link Sublayer.

The access network shall initialize the receive pointer, $V(R)$, to the MessageSequence field of the first PilotReport message it received in the Idle State, and the access network shall subsequently set it to the MessageSequence field of each received PilotReport message.

When the access network receives a PilotReport message in the Connected State, it shall validate the message using the procedure defined in 15.1.7. The access network shall discard the message if it is invalid.

11.7.4.4.2 MACResourceAssignment message validation

The access network shall set the MessageSequence field of the first MACResourceAssignment message it sends for the connection to zero. Subsequently, each time the access network sends a new MACResourceAssignment message in the Connected State, it shall increment this field. If the access network is sending the same message multiple times, it shall not change the value of this field between transmissions.³⁹

The access terminal shall initialize the receive pointer, $V(R)$, to the MessageSequence field of the first MACResourceAssignment message that it receives in the Connected State.

When the access terminal receives an MACResourceAssignment message in the Connected State, it shall validate the message using the procedure defined in 15.1.7. The access terminal shall discard the message if it is invalid.

11.7.4.4.3 SupplementalConfigAssignment message validation

The access network shall set the MessageSequence field of the first SupplementalConfigAssignment message it sends for the connection to zero. Subsequently, each time the access network sends a new SupplementalConfigAssignment message in the Connected State, it shall increment this field. If the access network is sending the same message multiple times, it shall not change the value of this field between transmissions.

The access terminal shall initialize the receive pointer, $V(R)$, to the MessageSequence field of the first SupplementalConfigAssignment message that it receives in the Connected State.

When the access terminal receives an SupplementalConfigAssignment message in the Connected State, it shall validate the message using the procedure defined in this subclause. The access terminal shall discard the message if it is invalid.

11.7.4.4.4 OFDMControlAssignment Message Validation

The access network shall set the MessageSequence field of the first OFDMControlAssignment message it sends for the connection to zero. Subsequently, each time the access network sends a new OFDMControlAssignment message in the Connected State, it shall increment this field. If the access network is sending the same message multiple times, it shall not change the value of this field between transmissions.

The access terminal shall initialize the receive pointer, $V(R)$, to the MessageSequence field of the first OFDMControlAssignment message that it receives in the Connected State.

When the access terminal receives an OFDMControlAssignment message in the Connected State, it shall validate the message using the procedure defined in 15.1.7. The access terminal shall discard the message if it is invalid.

³⁹The access network may send a message multiple times to increase its delivery probability.

11.7.4.4.5 CDMA Traffic Assignment Message Validation

The access network shall set the MessageSequence field of the first CDMA Traffic Assignment message it sends for the connection to zero. Subsequently, each time the access network sends a new CDMA Traffic Assignment message in the Connected State, it shall increment this field. If the access network is sending the same message multiple times, it shall not change the value of this field between transmissions.

The access terminal shall initialize the receive pointer, $V(R)$, to the MessageSequence field of the first CDMA Traffic Assignment message that it receives in the Connected State.

When the access terminal receives an CDMA Traffic Assignment message in the Connected State, it shall validate the message using the procedure defined in 15.1.7. The access terminal shall discard the message if it is invalid.

11.7.4.5 Inactive State

Upon entering this state, the access terminal shall perform the following:

- The access terminal shall set the Active Set, and the Candidate Set to NULL.
- The access terminal shall initialize the Remaining Set to contain all of the pilots.

11.7.4.6 Idle State

In the Idle State, the access terminal maintains an Active Set of size one and performs procedures related to access related indications.

11.7.4.6.1 Active Set maintenance

The access network shall not initially maintain an Active Set for the access terminal in this state.

The access terminal shall initially keep an Active Set of size one when it is in the Idle State. The Active Set pilot shall be the pilot associated with the Control Channel that the access terminal is currently monitoring. The access terminal shall return an *IdleHandoff* indication when the Active Set changes in the Idle State.

The access terminal shall not change its Active Set pilot at a time that causes it to miss a QuickPageBlock. Other rules governing when to replace this Active Set pilot are beyond the scope of this standard.

11.7.4.6.2 Processing access-related indications

The following operation shall be performed at the access terminal if the *AccessChannelMAC.AccessGrantReceived* Indication is received, and at the access network, if the *AccessChannelMAC.AccessProbeReceived* indication is received.

If the Idle State Protocol is in the Access State, the Active Set Management Protocol at the access terminal shall return a *ConnectionOpened* indication. The access terminal shall use the default fields in the MACResourceAssignment SupplementalConfigAssignment messages until it receives a SupplementalConfigAssignment message. block and place it in the public data. The fields in the block shall be decided based on the public data of the Overhead Messages Protocol, the received access grant, or the InitialSetupConfigurationAttribute. The access terminal shall use the default fields in the SupplementalConfigAssignment message until it receives a SupplementalConfigAssignment message.

The Active Set Management Protocol at the access network shall return a *ConnectionOpened* indication with *ConnectType* = InitialAccess. The access network shall use the default fields in the

MACResourceAssignment and SupplementalConfigAssignment messages until it sends a SupplementalConfigAssignment message.

11.7.4.7 Connected State

In this state, PilotReport messages from the access terminal are based on changes in the radio link between the access terminal and the access network, obtained through pilot strength measurements at the access terminal.

The access network determines the contents of the Active Set through MACResourceAssignment messages.

If the access terminal receives an *AirLinkManagement.IdleInitiated* indication, the access terminal shall transition to the Idle State.

11.7.4.7.1 Active Set maintenance

11.7.4.7.1.1 Access network

Whenever the access network sends an MACResourceAssignment message to the access terminal, it shall add to the Active Set any pilots listed in the message that are not currently in the Active Set. The access network shall place the most recently transmitted MACResourceAssignment message in the public data of the Active Set Management Protocol.

The access network shall delete a pilot from the Active Set if either of the following conditions is satisfied:

- The pilot was not listed in a MACResourceAssignment message and if the access network received the MACResourceAck message, acknowledging that MACResourceAssignment message.
- The access network received a PilotReport with the Keep field equal to zero for a pilot.

The access network should send an MACResourceAssignment message to the access terminal in response to changing radio link conditions, as reported in the access terminal's PilotReport messages.

The access network should only specify a pilot in the MACResourceAssignment message if it has allocated the required resources in the associated sector. This means that the sector specified by the pilot is ready to receive data from the access terminal and is ready to transmit queued data to the access terminal should the access terminal directs a handoff request to that sector.

If the access network receives a PilotReport with the Keep field equal to '1' for a pilot, the access network should send a MACResourceAssignment message adding that pilot to the Active Set. If the last received PilotReport message does not list a pilot, or has the Keep field equal to '0' for a pilot, then the access network shall not add that pilot to the Active Set.

If the access network adds or deletes a pilot in the Active Set, it shall send an MACResourceUpdated indication. If the access network adds a pilot to an Active Set when the Air Link Management Protocol is not in Connected State, the access network shall send a ConnectionOpened indication with ConnectType = ActiveSetAdd. If the Active Set at the access network becomes empty, the access network shall return an EmptyMACResourceAssigned indication.

11.7.4.7.1.2 Access terminal

If the access terminal receives a valid MACResourceAssignment message (see 11.7.4.4.2), it shall replace the contents of its current Active Set with the pilots specified in the message. The access terminal shall process the message as defined in 11.7.4.7.5.

If the access terminal receives a valid MACResourceAssignment message that causes the size of the Combined Active Set to exceed MaxSupportedActiveSetSize, the access terminal shall perform the following:

- Remove as many of the weakest pilots from the Active Set as required to make the Combined Active Set size equal to MaxSupportedActiveSetSize. Pilot A is considered weaker than pilot B:
 - If pilot A has an active drop timer but pilot B does not.
 - If both pilots have an active drop timer and pilot A's drop timer is closer to expiration than pilot B's.
 - If neither of the pilots has an active drop timer and pilot A's strength is less than pilot B's.
- Send a PilotReport message

If the access terminal sends a PilotReport message with the Keep field equal to '0' for a certain pilot (as specified in 11.7.5.1), and that pilot is in the Active Set, the access terminal shall remove that pilot from the Active Set.

The access terminal shall return an EmptyMACResourceAssigned indication if

- The access terminal sends a PilotReport with Keep=0 for all pilots in the Active Set of this InUse instance.
- The access terminal receives a valid MACResourceAssignment message with NumPilots field equal to zero.

11.7.4.7.2 ResetReport message

The access network may send a ResetReport message to reset the conditions under which PilotReport messages are sent from the access terminal. Access terminal usage of the ResetReport message is specified in the following subclause.

11.7.4.7.3 Pilot Strength Report Rules

The access terminal sends a PilotReport message to the access network in this state to request addition or deletion of pilots from its Active Set. The access terminal shall include in the PilotReport message the PilotID, pilot strength, and drop timer status for exactly one pilot in each Pilot Group in the Active Set and Candidate Set.

The access terminal shall send a PilotReport message if any one of the following occurs:

- Upon entering the Connected State.
- The access terminal receives a PilotReportRequest message. The access terminal shall set the TimingInfoIncluded field of the PilotReport message to 1 if it receives a PilotReportRequest message with the ReportFormat set to 0x01. If the PilotReportRequest message included any pilots, the access terminal shall include those pilots in the PilotReport.

If a pilot satisfies any of the following three conditions the access terminal shall send a PilotReport message. If a pilot satisfies any of the following three conditions and the pilot is not known to belong to any of the InUse instances, as determined by the AN Groups public data of the Overhead Messages Protocol, the access terminal shall generate a *RouteControl.CreateNewStack(PilotID)* command.

- The pilot is in the Remaining Set, and has pilot strength greater than the value specified by PilotAdd,
- The pilot is in the Candidate Set, and has pilot strength greater than the value specified by PilotCompare above an Active Set pilot, and a PilotReport message carrying this information has not been sent since the last ResetReport message was received.
- The pilot is in the Candidate Set, and has pilot strength above PilotAdd, and a PilotReport message carrying this information has not been sent since the last ResetReport message was received.

The access terminal shall send a PilotReport message if a pilot satisfies the following condition:

- The pilot is in the Active Set, and the drop timer of the pilot has just expired.

11.7.4.7.4 VCQI Report rules

The access terminal sends the VCQIReport message (VCQI refers to Vector Channel Quality Indicator) to convey extra channel quality information to the access network, beyond the information that is carried in the CQI channel defined by the physical layer.

If the VCQIReportingEnabled is equal to 0, the access terminal shall set NumPilots to 0 in all VCQIReport messages.

If VCQIReportingEnabled is equal to 1,

- The access terminal shall send a VCQIReport when it receives an *ActiveSetManagement.MACResourceUpdated* indication from any InUse instance.
- the access terminal shall send at least one VCQIReport with a non-zero NumPilots field every MaxVCQIReportInterval, and shall not send two VCQI reports with a non-zero NumPilots field in an interval less than MinVCQIReportInterval.

If CellNullReportingEnabled is equal to 1, the access terminal shall set CellNullCQIIncluded to 1 in all VCQIReport messages, and the access terminal shall send at least one VCQI message every CellNullCQIReportInterval superframes.

11.7.4.7.5 Processing the MACResourceAssignment message

The access terminal shall process a valid MACResourceAssignment message (see 11.7.4.4.2) as follows:

- The access terminal shall return a *MACResourceUpdated* indication.
- If the Air Link Management Protocol is not in Connected State, and the received MACResourceAssignment message has a nonzero NumPilots field, then the access terminal shall return a *ConnectionOpened* indication.
- The access terminal shall update its Active Set as defined in 11.7.4.7.1.2.
- The access terminal shall tune to the frequency defined by the ChannelBand record, if this record is included in the message. If more than one ChannelBand record is specified in the message, the access terminal may autonomously tune to one of the frequencies defined by the ChannelBand.
- The access terminal shall inform the physical layer about the synchronous status of sectors in the Combined Active Set according to the following rules—Two sectors shall be said to be mutually synchronous if any of the following conditions are met, and shall be said to be not mutually synchronous; otherwise
 - Both sectors are listed in the same MACResourceAssignment message and have the same non-zero SynchronousGroupID field.
 - Both sectors have the IsSynchronous field equal to 1 in the AcqInfo block of the Overhead Messages Protocol, and the Physical Layer has determined them to have the same Cyclic Prefix duration.
 - In all other cases the access terminal shall inform the physical layer that the two sectors are not mutually synchronous.
- The access terminal shall place the MACResourceAssignment message in the public data. If the MACResourceAssignment message has a non-zero value for the AdditionalFieldsStatus field for some pilot, the access terminal shall fill the additional fields (as defined in the MACResourceAssignment message) according to rules followed by the access network for setting the AdditionalFieldsStatus field (as defined in the MACResourceAssignment message).

- The access terminal shall send the access network an *MACResourceAck* message specifying the *MessageSequence* value received in the *MACResourceAssignment* message.

11.7.4.7.6 Processing the SupplementalConfigAssignment, CDMATrafficAssignment, and OFDMControlAssignment Messages

The access terminal shall process a valid message as follows:

- The access terminal shall return a *MACResourceUpdated* indication.
- The access terminal shall add the received message to the public data.

11.7.4.7.7 Rules for Transmission of the MACResourceStatus message

If the access terminal receives a *MACResourceUpdated* indication from any other *InUse* instance, the access terminal may transmit a *MACResourceStatus* message. The exact rules for the transmission of the *MACResourceStatus* message, are outside the scope of the specification. An access network may use the *MACResourceStatus* message to align its assigned MAC resources with MAC resources assigned to the access terminal by other access networks.

The access terminal shall send a *MACResourceStatus* message if this *InUse* instance is newly created to add a pilot to the Combined Active Set.

11.7.4.7.8 Processing the MACResourceAck message

The access network should set a transaction timer when it sends an *MACResourceAssignment* message. If the access network sets a transaction timer, it shall reset the timer when it receives an *MACResourceAck* message containing a *MessageSequence* field equal to the one sent in the *MACResourceAssignment* message.

If the timer expires, the access network should return an *ActiveSetManagement.ConnectionLost* indication.

11.7.4.7.9 Processing the SectorIDRequest message

If the access terminal receives a *SectorIDRequest* message, and has received a *SectorParameters* message from a sector with the *PilotID* indicated in the *SectorIDRequest* message within *SectorIDRequestMemoryCounter* previously visited sectors, then the access terminal shall respond with a *SectorIDResponse* message.

Otherwise, the access terminal shall ignore the *SectorIDRequest* message.

11.7.4.7.10 Procedures for MultiChannel Active Set

The *MACResourceAssignment* message can assign the access terminal an Active Set that includes pilots on more than one *ChannelBand*. The number of *ChannelBands* included in the Active Set by the access network shall not exceed the *MaxActiveSetCarriers* attribute. Given access terminal capability, only some of these assigned *ChannelBands* in the Combined Active Set can be used at a given time to transmit control and traffic channels.

If the access terminal receives a *ChannelSwitch* message, the access terminal shall perform handoff to the sector identified in the received message. The access terminal shall not initiate handoff request procedures between different *ChannelBands* within the same *PilotGroup*. For example, if an access terminal with a capability to receive one *ChannelBand* has in its Active Set two members within a *PilotGroup*, then the access terminal shall not initiate handoff request procedures between these two members of the Active Set by itself. The access terminal shall not initiate handoff request procedures between pilots with the same

PilotID that belong to different PilotGroups if the serving pilot's PilotStrength is greater than ATChannelSwitchThreshold.

If the access terminal is not transmitting reverse control channels to a pilot, then this protocol shall set the AccessBasedHandoffRequired public data associated with the pilot to 1. If the last received MACResourceAssignment message has the ReverseControlChannelEnabled field equal to zero for a pilot, the access terminal shall set AccessBasedHandoffRequired public data associated with the pilot to 1. The access terminal shall set AccessBasedHandoffRequired for all other pilots to 0.

If the last received MACResourceAssignment message has the ReverseControlChannelEnabled field equal to zero from a pilot that is not a serving sector, then the access terminal shall stop transmitting reverse control channels to that pilot. If the last received MACResourceAssignment message has the ReverseControlChannelEnabled field equal to zero for a pilot that is a forward or reverse serving sector then the access terminal shall continue transmission of reverse control channels to that pilot.

If due to capability limitation or received ReverseControlChannelEnabled fields, the access terminal is unable to transmit control channels to a member of the active set, and the forward link serving sector belongs to a different ANGroup, then the access terminal shall send a MACResourceStatus message that indicates the set of active set members to which the access terminal is transmitting reverse control channels. Upon receiving this message, sector to which the access terminal is not transmitting any reverse control channels may stop demodulation of the reverse control channels.

11.7.5 Message formats

11.7.5.1 PilotReport

The access terminal sends the PilotReport message to notify the access network of its current location and provide it with an estimate of its surrounding radio link conditions. The information in this message is used by the access network to add or remove sectors from the Active Set (see Table 300).

Table 300—PilotReport message

Field	Length (bits)
MessageID	8
MessageSequence	8
ReferencePilotID	10
ReferencePilotStrength	6
ReferenceKeep	1
NumPilots	6

NumPilots occurrences of the following record:

PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
PilotStrength	6
Keep	1
ChannelInfoRequested	1

Table 300—PilotReport message (continued)

Field	Length (bits)
TimingInfoIncluded	1
SubSuperframeOffset	0 or 16
SuperframeOffset	0 or 16
Padding	Variable

MessageID	The access terminal shall set this field to 0x00.
MessageSequence	The access terminal shall set this field to the sequence number of this message. The sequence number of this message is 1 more than the sequence number of the last PilotReport message (modulo 256) sent by this access terminal. If this is the first PilotReport message sent by the access terminal, it shall set this field to 0x00.
ReferencePilotID	The access terminal shall set this field to the access terminal's time reference (the reference pilot).
ReferencePilotStrength	The access terminal shall set this field to $\lfloor -2 \times 10 \times \log_{10} PS \rfloor$, where PS is the strength of the reference pilot, measured as specified in 11.7.4.3.1. If this value is less than 0, the access terminal shall set this field to '000000'. If this value is greater than '111111', the access terminal shall set this field to '111111'.
ReferenceKeep	If the pilot drop timer corresponding to the reference pilot has expired, the access terminal shall set this field to '0'; otherwise, the access terminal shall set this field to '1'. If the access terminal sets this field to '1', the access network should add this pilot to the Active Set. If the access terminal sets this field to '0', the access network should remove this pilot from the Active Set.
NumPilots	The access terminal shall set this field to the number of pilots that follow this field in the message.
PilotID	The PilotID of a pilot in the Active Set or Candidate Set of the access terminal that is not the reference pilot
ChannelBandIncluded	The access terminal shall set this field to '1' if the channel for this pilot offset is not the same as the current channel. Otherwise, the access terminal shall set this field to '0'.
ChannelBand	The access terminal shall include this field if the ChannelBandIncluded field is set to '1'. The access terminal shall set this to the ChannelBand record (see 15.2.1) corresponding to this pilot. Otherwise, the access terminal shall omit this field for this pilot offset.
PilotStrength	The access terminal shall set this field to $\lfloor -2 \times 10 \times \log_{10} PS \rfloor$, where PS is the strength of the pilot (PilotStrength) in the above field, measured as specified in 11.7.4.3.1. If this value is less than 0, the access terminal shall set this field to '000000'. If this value is greater than '111111', the access terminal shall set this field to '111111'.

Keep	The access terminal shall set this field according to the pilot rules specified in 11.7.4.7.1.2. A value of '1' indicates that this pilot is already in the Active Set. A value of '0' indicates that the access terminal has removed this pilot from the Active Set.
ChannelInfoRequested	If this sector is not part of the Active Set, and ChannelInfoUpToDate=0 for this sector, where ChannelInfoUpToDate is defined by the Overhead Messages Protocol, the access terminal shall set this field to '1'. Otherwise the access terminal shall set this field to '0'. If the access network adds a sector to the Active Set, and ChannelInfoRequested=0 in the received PilotReport, the access network shall send overhead messages to the access terminal.
TimingInfoIncluded	The setting for this field is described in the rules for transmission of a PilotReport message.
SubSuperframeOffset	This field shall be included if TimingInfoIncluded is set to '1' and SubSuperframeOffset is not equal to all ones. If included, this field shall be set to the value $2^{15} + m - n$, where m and n are described in the discussion of the SubSuperframeOffset. The all ones value of this field corresponds to an offset that is not expected to be seen by the access terminal, and reserving this field does not lead to a loss of accuracy.
SuperframeOffset	This field shall be included if TimingInfoIncluded is set to '1'. This field shall be set to the value $2^{15} + m - n$, where m and n are described in the discussion of the SubSuperframeTimingOffset. The receiver shall ignore this field if SubSuperframeOffset is equal to the all ones value.
Padding	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros. The receiver shall ignore this field.

Channels	RTC
Addressing	Unicast

RLP	Best effort
AuthTag	Required when key is available

11.7.5.2 VCQIReport

The access terminal shall send this message to report VCQI (see Table 301).

Table 301—VCQIReport message

Field	Length (bits)
MessageID	8
CellNullCQIIncluded	1
CellNullCQI	0 or 4
ReportType	1
NumPilots	6

NumPilots Occurrences of the following record{

Table 301—VCQIReport message (continued)

Field	Length (bits)
PilotID	10
ChannelBandSameAsPrevious	1
ChannelBand	Variable
NumResourceSets	5
NumResourceSet instances of the following record{	
ResourceSetID	5
NumRanksOrLayers	4
NumRanksOrLayers occurrences of the following record{	
VCQI	4
}}}	
Padding	Variable

MessageID	The access terminal shall set this field to 0x01.
CellNullCQIIncluded	If a CellNullCQI value is not included the access terminal shall set this field to 0. Otherwise, the access terminal shall set this field to 1.
CellNullCQI	The access terminal shall include this field if CellNullCQIIncluded is 1, and shall set this field to the CellNullCQI defined for the forward link serving sector in the Reverse Control Channel MAC. Otherwise, the access terminal shall omit this field.
ReportType	This field shall be set to 0 for SCW and 1 for MCW report.
NumPilots	The access terminal shall set this field to the number of sectors for which the report is being sent. The access terminal should include all members of the Combined Active Set in this message.
PilotID	The access terminal shall set this field to a PilotID of a pilot in Active Set.
ChannelBandSameAsPrevious	The access terminal shall set this field to '1' if there is only one ChannelBand in the Combined Active Set, or if this ChannelBand is the same as the ChannelBand for the previous pilot listed in this message. The access terminal shall set this field to '0' otherwise.
ChannelBand	This field shall be included if ChannelBandSameAsPrevious is equal to '0', and shall be omitted otherwise. If included, this field shall be set to the ChannelBand record (see 15.2.1) for this pilot.
NumResourceSets	The access terminal shall set this field to the number of resource sets being reported.

ResourceSetID	The access terminal shall set this field to the ResourceSetID corresponding to the following record.
NumRanksOrLayers	The number of ranks or layers reported in the message.
VCQI	If ReportType is 0, this field shall be set to the VCQIValueSCW for this ResourceSet and Rank. If ReportType is 1, this field shall be set to the VCQIValueMCW for this ResourceSet and Layer. VCQIValueSCW and VCQIValueMCW are defined in the Reverse Control Channel MAC Protocol.
Padding	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros. The receiver shall ignore this field.

Channels	RTC

RLP	Best effort

Addressing	Unicast

AuthTag	Required when key is available

11.7.5.3 SupplementalConfigAssignment

This message shall be sent by the access network to assign reverse control channel parameters and VCQI reporting parameters to the access terminal. The default values listed in the message definition below are not part of the message. The access terminal shall use these default values before it has received this message for any assigned MACID (see Table 302).

Table 302—SupplementalConfigAssignment message

Field	Length (bits)	Default Value
MessageID	8	N/A
MessageSequence	8	N/A
NumPilots	6	N/A
NumPilots instances of the following record {		
PilotId	10	N/A
ChannelBandIncluded	1	0
ChannelBand	Variable	N/A
BFCHCodebookType	1	0
BFCHBeamCodeBookIndex	1 or 16	0
MaxNonFLSSCQIReportRate	3	010
MaxPAHCHReportRate	3	111
MaxPSDReportRate	3	111
MinRequestInterval	3	001
NumSecondaryRPICHSubsegments	2	0

NumSecondaryRPICHSubsegments occurrences of the following field:

Table 302—SupplementalConfigAssignment message (continued)

Field	Length (bits)	Default Value
SecondaryRPICHSubsegmentj	6	N/A
NumSFCHSubzones	3	011
VCQIReportingEnabled	1	0
MaxVCQIReportInterval	8 ^a	0
MinVCQIReportInterval	8 ^a	0
VCQIMeasureInterval	2 ^a	0
NumResourceSets	5	0

NumResourceSets instances of the following record{

ResourceSetBitmap	8	N/A
ResourceSetBRCHSubzoneCyclingEnabled	1	N/A
ResourceSetSubZoneSpacing	2	N/A
NumResourceSubzones	5	N/A
ResourceSetOffset	5	N/A

}

NumSubbands	5	00000

NumSubbands occurrences of the following field

SubBandCQIOffsetj	4	100

}

Padding	0–7	

^aThis field may be absent (see field description for conditions).

MessageID	The access network shall set this field to 0x02.
MessageSequence	The access network shall set this field to 1 higher than the MessageSequence field of the last SupplementalConfigAssignment message (modulo 2 ^S , S=8) sent to this access terminal.
NumPilots	This field shall be set to the number of active set members for which information is included in this message.
PilotID	This field shall be set to the PilotID of the sector to which this message is directed.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.

ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.
BFCHCodebookType	This field shall be set to 0 if the BFCHCodebookIndex has length 1 bit, and shall be set to 1 if BFCHCodebookIndex has length 16 bits.
BFCHBeamCodeBookIndex	This field shall be set to the BFCHBeamCodeBookIndex. The one bit value 0 shall correspond to a binary codebook; 1 shall correspond to Fourier based codebook. A sixteen bit value shall correspond to the CodebookID of a downloadable codebook, as defined in the Forward Traffic Channel MAC Protocol.
MaxNonFLSSCQIReportRate	This field shall specify the maximum number of CQI reports the access terminal may send to non FLSS sectors over eight superframes.
MaxPAHCHReportRate	This field shall specify the maximum number of reports the access terminal shall send over thirty two r-pach opportunities. This parameter shall take value 2^n , where n is the value of the three bit field.
MaxPSDReportRate	This field shall specify the maximum number of reports the access terminal shall send over thirty two r-psdch opportunities. This parameter shall take value 2^n , where n is the value of the three bit field.
MinRequestInterval	The access network shall set this field to one less than the gap (in units of 4 PHY frames) between two request transmissions by the access terminal.
NumSecondaryRPICHSubsegments	The number of secondary RPICH Subsegments.
SecondaryRPICHSubsegmentj	This field shall be set to identify the location of the j^{th} SecondaryRPICHSubSegment.
NumSFCHSubzones	The number of SHCH Subzones.
VCQIReportingEnabled	The access terminal shall disable VCQI reports if this parameter is equal to 0; otherwise the access terminal shall send VCQI reports according to the procedures defined in this protocol.
MaxVCQIReportInterval	This field shall be included if VCQIReportingEnabled is '1', and shall be omitted otherwise. This parameter shall take the value $1+4n$ superframes.
MinVCQIReportInterval	This field shall be included if VCQIReportingEnabled is '1', and shall be omitted otherwise. This parameter shall take the value $1+4n$ superframes.
VCQIMeasureInterval	The VCQI measurement interval in units of superframes, encoded as $1+2n$, where n is the value of this field.
NumResourceSets	The number of resource sets on this sector.

ResourceSetBitmap The j^{th} bit of this field shall be set to 1 if a frame with frame index mod 8 = j contains a subzone that corresponds to this resource set.

ResourceSetBRCHSubzoneCyclingEnabled

This field shall be set to 1 if BRCHSubzoneCycling is enabled on this sector. For BRCH resource set with BRCHSubzoneCycling disabled or DRCH resource set, the first hop-port subzone offset on all interlaces where this resource set is present shall be set to the ResourceSubzoneOffset. For BRCH resource set with BRCHSubzoneCycling enabled, the offset of the first hop-port subzone over each interlace is shifted cyclically. Since the offset of first hop-port subzone over the lowest indexed interlace is defined by ResourceSubzoneOffset, the offset of the first hop-port subzone in the next interlace, where the resource set is present, is increased by 1 mod NumBRCHSubzones.

ResourceSetSubZoneSpacing

This field shall indicate the spacing between subzones in a resource set. Subzones belonging to a resource group on an interlace shall be equally spaced, where the first hop-port subzone is defined by ResourceSubzoneOffset and ResourceSetBRCHSubzoneCyclingEnabled.

NumResourceSubzones This field shall determine the number of subzones in each interlace where the resource set is present. An interlace is defined as the set of frames that have the same value of Frame Index mod 8. This parameter takes the value ' $n + 1$ '.

ResourceSubzoneOffset This field shall be set to the first subzone on the lowest indexed interlace that is part of a resource set. Interlace index i is defined as the interlace containing the set of frames that have Frame Index mod 8 = i .

NumSubbands This field shall determine the number of subbands. This parameter shall take the value ' $n + 1$ ', where n is the value of this 5-bit field.

SubBandCQIOffset j This parameter shall take the value $-8 + n$ dB.

Padding The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros. The access terminal shall ignore this field.

Channels	FTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

11.7.5.4 MACResourceAssignment

The access network sends the MACResourceAssignment message to manage the access terminal's Active Set (see Table 303).

Table 303—MACResourceAssignment message

Field	Length (bits)	Default value
MessageID	8	N/A
MessageSequence	8	N/A

Table 303—MACResourceAssignment message (continued)

Field	Length (bits)	Default value
SchedulerGroupIDIncluded	1	0
PilotGroupIDIncluded	1	0
CellGroupIDIncluded	1	0
NumPilots	4	1

NumPilots occurrences of the following record{

PilotID	10	Pilot PN of sector that sent Access Grant
ActiveSetIndex	6	0
ChannelBandSameAsPrevious	1	0
ChannelBand	Variable	ChannelBand where access was made
ReverseControlChannelRequired	1	1
AdditionalFieldsStatus	1	1
The following fields shall be included if AdditionalFieldsStatus is equal to 1 {		
MACID	11	MAC ID Assigned at access
SchedulerGroupID	6 ^a	N/A
PilotGroupID	3 ^a	N/A
CellGroupID	3 [*]	N/A
SynchronousGroupID	3 [*]	N/A
SingleServingSector	1	1
StartofPacketMode	1	0
StartofPacketFrame	2	N/A
NumFailPacketsforDrop	2	0
HalfDuplexMode	1	Depends on AT Capability
ROFDMEnabledFL	1	0
SCCHTailBitingCodeEnabled	1	0
LDPCSupportedRL	1	0
LDPCSupportedFL	1	0
CDCCHSubsegmentIndex	6	MAC ID mod Sum(NumCDMA Subsegmentsj)
CDCCHPeriod	3	000
CDCCHPhase	3	000

Table 303—MACResourceAssignment message (continued)

Field	Length (bits)	Default value
CQIPilotReportInterval	3	000
CQIPilotReportPhase	3	000
CQIReportInterval	3	000
CQIReportPhase	3	000
CQIreportRate	3	000
UserPQIPeriod	2	00
NumPrimaryRPICHSubsegments	2	1
NumPrimaryRPICHSubsegments of the following field		
PrimaryRPICHSubsegment j	6	MAC ID mod Sum(NumCDMA Subsegments j)
PrimaryRPICHPeriod	3	1
PrimaryRPICHPhase	3	0
SHOGIncluded	1	0
The following record shall be included if SHOGIncluded=1 {		
NumSHOG	3	
NumSHOG Instances of the following record{		
SHOGMasterActiveSetIndex	6	N/A
SHOG_ID	2	N/A
NumSHOGSlaves	2	N/A
NumSHOGSlaves instances of the following field{		
SHOGSlaveActiveSetIndex	6	N/A
}}}		
Padding	Variable	N/A

^aThis field may be absent (see field description for conditions).

MessageID	The access network shall set this field to 0x03.
MessageSequence	The access network shall set this field to 1 higher than the MessageSequence field of the last MACResourceAssignment message (modulo 2^S , $S = 8$) sent to this access terminal.
SchedulerGroupIDIncluded	This field shall determine whether the SchedulerGroupID is included for the Pilots contained in this message.
PilotGroupIDIncluded	The access network shall set this field to '1' if PilotGroupID fields are included in this message. Otherwise, the access network shall set this field to '0'.

CellGroupIDIncluded	The access network shall set this field to '1' if CellGroupID fields are included in this message. Otherwise, the access network shall set this field to '0'.
NumPilots	The access network shall set this field to the number of pilots included in this message.
PilotID	The access network shall set this field to the PilotID of a pilot included in the message.
ActiveSetIndex	This field shall be set to a short identifier for this member of the active set. The access terminal shall purge an ActiveSetIndex value if it receives this message with the corresponding ActiveSetIndex value omitted.
ChannelBandSameAsPrevious	If this is the first pilot listed in the message, the access network shall set this field to '0'. The access network shall set this field to '1' if the ChannelBand record for this pilot is the same as the ChannelBand record for the previous pilot. The access network shall set this field to '0' if the ChannelBand for this Pilot is different from the ChannelBand of the previous Pilot.
ChannelBand	If the ChannelBandSameAsPrevious field is set to '0' the access network shall omit this field. The access network shall include this field if the ChannelBand SameAsPrevious field is set to '1'. If included, this field shall be set to the ChannelBand record (see 15.2.1) for this pilot.
ReverseControlChannelRequired	The access network shall set this field to 1 to require the access terminal to transmit reverse control channels to this sector. The access network shall set this field to 0 to indicate the access terminal is not required to transmit reverse control channels to this sector when this sector is not the serving sector. If this bit is set to zero, the access terminal shall use access based handoff to handoff to this sector.
AdditionalFieldsStatus	This field shall describe the status of the additional fields described in the message definition. AdditionalFieldsStatus shall be set to '0' if the additional fields are not included in the message, and the values of the additional fields for this sector are identical to the values for the same pilot in the last transmitted MACResourceAssignment message. AdditionalFieldsStatus shall be set to '1' if the additional fields are included in the message.
MAC ID	The access network shall set this field to the MAC ID assigned to the user in this sector.
SchedulerGroupID	The access network shall include this field if the SchedulerGroupIDIncluded field is set to '1', and shall exclude this field otherwise. If included, the access network shall set this field to a number that indicates the Scheduler Group to which this pilot belongs.
PilotGroupID	The access network shall omit this field if PilotGroupIDIncluded is set to '0'. If PilotGroupIDIncluded is set to '1', the access network shall set this field to the PilotGroupID associated with the neighbor pilot.
CellGroupID	The access network shall omit this field if CellIDIncluded is set to '0'. If CellGroupIDIncluded is set to '1', the access network shall set this field to the

	CellGroupID associated with the pilot. The CellGroupID fields carry information about which sectors included in this message belong to the same cell, and the fields values by themselves have no meaning across different occurrences of this message. The access network shall assign the same CellGroupID to two sectors included in this message if the two sectors belong to the same cell. The access network shall assign different CellGroupID to two sectors included in this message if the two sectors belong to different cells.
SynchronousGroupID	This field shall be included if the MSB of this PilotID is equal to 0, and shall be omitted otherwise. The SynchronousGroupID fields carry information about timing synchronization of sectors included in this message, and the field values by themselves have no meaning outside this message. The access network shall assign the same SynchronousGroupID to two sectors included in this message if the two sectors are mutually synchronous. The access network shall assign different SynchronousGroupID fields to two sectors included in this message if the two sectors are not mutually synchronous. The SynchronousGroupID value of zero shall be used only for sectors about which synchronization information is not known.
SingleServingSector	The access network shall set this field to '1' if the access terminal is required to maintain the same forward and reverse link serving sectors. The access network shall set this field to '0' otherwise. The access network should set this field to '1' if the DisjointLinkSupported attribute for the terminal is equal to '0'.
StartofPacketMode	This field determines when the access terminal begins decoding a packet corresponding to a given assignment. If StartofPacketMode is set to '0', then the access terminal begins decoding the next packet on the next assigned PHY frame for the assignment. Otherwise, if StartofPacketMode = 1, the time the access terminal starts decoding the next packet is determined by the StartofPacketFrame field.
StartofPacketFrame	This field shall be included if StartofPacketMode is equal to '1', and shall be omitted otherwise. If included, this field determines when the access terminal begins decoding a packet corresponding to a given assignment. If StartOfPacketMode is 1, the AT looks for a SOP every StartofPacketFrames frames (or rather transmissions). This field shall take values 6, 9, and 12 if the field is set to '00', '01', and '10' respectively. The value '11' is reserved.
NumFailPacketsforDrop	This field shall be set to one less than the number of packet errors that causes the access terminal to remove persistent assignment from the ATA.
HalfDuplexMode	This field shall be set to '0' if the access terminal is not assigned to operate in half-duplex mode. This field shall be set to '1' to indicate that the terminal is required to operate in half-duplex mode.
ROFDMEEnabledFL	This field shall be set to 1 if Rotational OFDM is enabled on the forward link. The access network shall set this field to '0' if the ROFDMSupportedFL attribute of the MAC Layer is equal to zero.
SCCHTailBitingCodeEnabled	This field determines if the access terminal is required to use tail-biting convolutional codes for Unicast transmissions on the F-SCCH. The access

network shall set this field to '0' if the SCCHTailBitingCodeSupported attribute of the MAC Layer is equal to zero.

LDPCSupportedRL The access network shall set this field to '1' to enable LDPC encoded transmissions on the reverse link for certain packet sizes. The access network shall set this field to '0' to disable all LDPC encoded transmissions on the reverse link.

LDPCSupportedFL The access network shall set this field to '1' to enable LDPC encoded transmissions on the forward link for certain packet sizes. The access network shall set this field to '0' to disable all LDPC encoded transmissions on the forward link.

CDCCHSubsegmentIndex This field shall be set to the CDMA subsegment on which the access terminal transmits R-CDCCH.

CDCCHPeriod This field determines the periodicity of R-CDCCH transmissions by the access terminal. This parameter takes the value $n + 1$.

CDCCHPhase This field determines the phase of R-CDCCH transmissions by the access terminal. This field shall be set to a value less than or equal to the CDCCHPeriod field.

CQIPilotReportInterval This field determines the periodicity at which an access terminal shall report pilot r-cqich reports on R-CDCCH. This field shall be set to '0' if the access terminal is not required to transmit pilot r-cqich reports. This parameter shall take the value $4 \times n$.

CQIPilotReportPhase This field determines the phase of the r-cqich reports. This field shall be set to a value less than CQIPilotReportInterval.

CQIReportInterval This field determines the periodicity at which the access terminal may report r-cqich on R-CDCCH. This parameter shall take the value $n + 1$.

CQIReportPhase This field shall determine the phase of the CQI reports that are transmitted with periodicity CQIReportInterval. This field shall be set to a value less than the CQIReportInterval parameter.

CQIReportRate This field shall determine the number of transmission of CQI reports over eight r-cqich opportunities. This parameter shall be set to one more than the value of this field.

UserPQIPeriod This field determines the periodicity of PQI reports transmitted by the access network. This parameter shall take the value 2^n .

NumPrimaryRPICHSubsegments
The number of primary RPICH Subsegments.

PrimaryRPICHSubsegment j
This field shall be set to identify the location of the j^{th} PrimaryRPICHSubsegment.

PrimaryRPICHPeriod	This field determines the period of the PrimaryRPICH. This parameter shall take the value $n + 1$.
PrimaryRPICHPhase	This field shall be set to the phase of the PrimaryRPICH.
SHOGIncluded	This field shall be determine if a softer handoff group is included in this message.
NumSHOG	This field shall be included if SHOGIncluded is equal to 1 and shall be omitted otherwise. If included, this field shall be set to the number of softer handoff groups specified in this message.
SHOGMasterActiveSetIndex	This field shall be included if SHOGIncluded is equal to 1 and shall be omitted otherwise. This field shall identify the active set member that is master in this softer handoff group.
SHOG_ID	This field shall be included if SHOGIncluded is equal to 1 and shall be omitted otherwise. The SHOGID of 0 shall be reserved, and may be used for control channel transmissions. This field shall provide an index of the softer handoff group.
NumSHOGSlaves	This field shall be included if SHOGIncluded is equal to 1 and shall be omitted otherwise. This field shall identify the number of active set members that are slaves in the softer handoff group.
SHOGSlaveActiveSetIndex	This field shall identify the active set members that are slaves in the softer handoff group.
Padding	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros. The access terminal shall ignore this field.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required when key is available

11.7.5.5 MACResourceAck

The access terminal sends the MACResourceAck message to provide an acknowledgment for the MACResourceAssignment message (see Table 304).

Table 304—MACResourceAck message

Field	Length (bits)
MessageID	8
MessageSequence	8

MessageID The access terminal shall set this field to 0x05.

MessageSequence The access terminal shall set this field to the MessageSequence field of the MACResourceAssignment message whose receipt this message is acknowledging.

Channels	RTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required when key is available

11.7.5.6 MACResourceStatus

The access terminal sends the MACResourceStatus message to assist different access networks coordinate the MAC resources assigned to the access terminal (see Table 305).

Table 305—MACResourceStatus message

Field	Length (bits)
MessageID	8
MessageSequence	8
RPICHScramblingSeed	40
NumPilots	6

NumPilots occurrences of the following record{

PilotID	10
ChannelBandSameAsPrevious	1
ChannelBand	Variable
HalfDuplexMode	1
MACID	11 ^a
ReverseControlChannelRequired	1
CQIStatus	1

The following fields shall be included if CQIStatus is equal to 1 and shall be omitted otherwise{

CQIPilotReportInterval	3
CQIPilotReportPhase	3
CQIReportInterval	3
CQIReportPhase	3
CDCCHSubsegmentIndex	6
CDCCHPeriod	3
CDCCHPhase	3
NumPrimaryRPICHSubsegments	2
NumPrimaryRPICHSubsegments of the following field	

Table 305—MACResourceStatus message (continued)

Field	Length (bits)
PrimaryRPICHSubsegment j	6
PrimaryRPICHPeriod j	3
PrimaryRPICHPhase j	3
}	
Padding	Variable

^aThis field may be absent (see field description for conditions).

MessageID	The access network shall set this field to 0x04.
MessageSequence	The access network shall set this field to 1 higher than the MessageSequence field of the last MACResourceStatus message (modulo 2^S , $S=8$) sent by the access terminal.
RPICHScramblingSeed	This field is public data of the Forward Link Control Segment MAC Protocol.
NumPilots	This access terminal shall set this field to the number of pilots included in this message. The access terminal shall include all pilots in the Active Sets of other InUse instances.
PilotID	The access terminal shall set this field to a PilotID of a pilot in Active Set of another InUse instance.
ChannelBandSameAsPrevious	The access terminal shall set this field to '1' if there is only one ChannelBand in the Combined Active Set and the Candidate Set, or if this ChannelBand is the same as the ChannelBand for the previous pilot listed in this message.
ChannelBand	This field shall be included if ChannelBandSameAsPrevious is equal to '0', and shall be omitted otherwise. If included, this field shall be set to the ChannelBand record (see 15.2.1) for this pilot.
HalfDuplexMode	The access terminal shall set this field to the HalfDuplexMode for this pilot.
MACID	This field shall be included if HalfDuplexMode is '1', and shall be omitted otherwise. The access terminal shall set this field to the MACID field assigned for this pilot.
ReverseControlChannelRequired	This field shall be set to the ReverseControlChannelRequired public data for this pilot.
CQIStatus	The access terminal shall set this field to 1 for the first sector included in this message. The access terminal shall set this field to 0 for a sector if the optional parameters following this field are the same as those for the first sector in the message. The access terminal shall set this field to 1 otherwise.

CQIPilotReportInterval	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the CQIPilotReportInterval field for this pilot.
CQIPilotReportPhase	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the CQIPilotReportPhase field for this pilot.
CQIReportInterval	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the CQIReportInterval field for this pilot.
CQIReportPhase	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the CQIReportPhase field for this pilot.
CDCCHSubsegmentIndex	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the CDCCHSubsegmentIndex field for this pilot.
CDCCHPeriod	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the CDCCHPeriod field for this pilot.
CDCCHPhase	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the CDCCHPhase field for this pilot.
NumPrimaryRPICHSubsegments	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the NumPrimaryRPICHSubsegments field for this pilot.
PrimaryRPICHSubsegmentj	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the PrimaryRPICHSubsegmentj field for this pilot.
PrimaryRPICHPeriodj	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the PrimaryRPICHPeriodj field for this pilot.
PrimaryRPICHPhasej	This field shall be omitted if CQIStatus is '0', and shall be included otherwise. If included, this field shall be set to the PrimaryRPICHPhasej field for this pilot.
Padding	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros. The access terminal shall ignore this field.

Channels	RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

11.7.5.7 OFDMControlAssignment

This message shall be sent by the access network to assign the terminal resources on the ODCCH (see Table 306).

Table 306—OFDMControlAssignment message

Field	Length (bits)
MessageID	8
MessageSequence	8
NumPilots	6
NumPilots instances of the following record {	
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
ODCCHResourcesIncluded	1
The following fields shall be included if ODCCHResourcesIncluded=1 {	
MIMOMode	1
MinCQICHReportRate	4
MinBFCHReportRate	4
MinMQICHReportRate	4
MinSFCHReportRate	4
NumODCCHReportIndices	2
NumODCCHReportIndices occurrences of the following	
ODCCHReportIndex _j	9
ODCTargetCtoI	3
ODCCHPeriod	3
ODCCHPhase	3
ODCCHReportRate	3
BFCHReportingMode	1
}}	
Padding	Variable

MessageID The access network shall set this field to 0x0c.

MessageSequence The access network shall set this field to 1 higher than the MessageSequence field of the last OFDMControlAssignment message (modulo 2^S , $S = 8$) sent to this access terminal.

NumPilots	This field shall be set to the number of active set members for which information is included in this message.
PilotID	This field shall be set to the PilotID of the sector to which this message is directed.
ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 5.2.1) for of the sector to which this message refers. This field shall be omitted otherwise.
ODCCHResourcesIncluded	This field shall be set to 1 if the access terminal is assigned resources on the ODCCH. This field shall be set to 0 otherwise.
MIMOMode	This field shall indicate SCW or MCW modes, when the field is set to 0 and 1 respectively.
MinCQICHReportRate	This field shall specify the minimum number of CQICH reports the access terminal shall send over eight R-ODCCH opportunities.
MinBFCHReportRate	This field shall specify the minimum number of BFCH reports the access terminal shall send over eight R-ODCCH opportunities.
MinMQICHReportRate	This field shall specify the minimum number of MCQI reports the access terminal shall send over eight R-ODCCH opportunities.
MinSFCHReportRate	This field shall specify the minimum number of SFCH reports the access terminal shall send over eight R-ODCCH opportunities.
NumODCCHReportIndices	This field shall determine the number of ODCCHReportIndices.
ODCCHReportIndex j	This field shall be set to ODCCHReportIndex j .
ODCCHPeriod	This field shall be set to one less than the periodicity of R-ODCCH in units of eight PHY frames.
ODCCHPhase	This field shall be set to the phase of R-ODCCH in units of eight PHY frames.
ODCCHReportRate	This field shall be set to one less than the number of reports the access terminal is required to send over eight ODCCH opportunities.
ODCTargetCtoI	This parameter shall take the value $-3 + n$, where n is the value of the three bit field.
BFCHReportingMode	This field shall be set to 0 for non-SDMA and 1 for SDMA reporting modes.

Padding The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros. The access terminal shall ignore this field.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required when key is available

11.7.5.8 11.7.5.8 CDMA Traffic Assignment

This message shall be sent by the access network to assign the terminal resources on the CDCH (see Table 307).

Table 307—CDMA Traffic Assignment message

Field	Length (bits)
MessageID	8
MessageSequence	8
NumPilots	6
NumPilots instances of the following record {	
ChannelBandIncluded	1
ChannelBand	Variable
NumCDCHSubsegments	4
NumCDCHSubsegments occurrences of the following fields {	
CDCHSubsegment j	6
PacketStartInterlace i , $i = 0, 1$	3
CDMAFACKIndex	8
}	
Padding	Variable

MessageID The access network shall set this field to 0x06.

MessageSequence The access network shall set this field to 1 higher than the MessageSequence field of the last CDMA Traffic Assignment message (modulo 2^S , $S=8$) sent to this access terminal.

NumPilots This field shall be set to the number of active set members for which information is included in this message.

PilotID This field shall be set to the PilotID of the sector to which this message is directed.

ChannelBandIncluded	This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.
ChannelBand	This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.
NumCDCHSubsegments	The number of CDCH subsegments assigned to the access terminal.
CDCHSubsegment j	The location of the j^{th} CDCH subsegment.
PacketStartInterlace i	The interlace on which the access terminal is allowed to start packet transmission
CDMAFACKIndex	This field determines the position of the forward acknowledgement channel for R-CDCH.
Padding	The number of bits in this field is equal to the number needed to make the message length an integer number of octets. This field shall be set to all zeros. The access terminal shall ignore this field.

Channels	RTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key available

11.7.5.9 ResetReport

The access network sends the ResetReport message to reset the PilotReport transmission rules at the access terminal (see Table 308).

Table 308—ResetReport message

Field	Length (bits)
MessageID	8

MessageID The access network shall set this field to 0x07.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key available

11.7.5.10 PilotReportRequest

The access network sends a PilotReportRequest message to request the access terminal to send a PilotReport message (see Table 309).

Table 309—PilotReportRequest message

Field	Length (bits)
MessageID	8

Table 309—PilotReportRequest message (continued)

Field	Length (bits)
ReportFormat	8
NumPilots	4
NumPilots occurrences of the following record{	
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
}	
Padding	Variable

MessageID	The access network shall set this field to 0x09.
ReportFormat	The access network shall set this field to indicate the format of the PilotReport it is requesting from the access terminal. The valid values for this field are 0x00 and 0x01, where 0x01 indicates that a detailed pilot report is requested.
NumPilots	This field shall be set to the number of pilots included in this message.
PilotID	This field shall be set to the PilotID of the pilot for which the access network is requesting a pilot strength
ChannelBandIncluded	This field shall determine if a ChannelBand is included for this pilot.
ChannelBand	If the ChannelBandIncluded field is set to '0' the access network shall omit this field. Otherwise, the access network shall include this field. This field shall be set to the ChannelBand record (see 15.2.1) of the Pilot for which the access network is requesting a pilot strength.
Padding	The length of this field shall be such that this message is octet aligned. The sender shall set this field to zero. The receiver shall ignore this field.

Channels	FTC
Addressing	Unicast

RLP	Best effort
AuthTag	Required if key available

11.7.5.11 SectorIDRequest

The access network sends a SectorIDRequest message to request the access terminal to send the SectorID of a given pilot (see Table 310).

Table 310—SectorIDRequest message

Field	Length (bits)
MessageID	8

Table 310—SectorIDRequest message (continued)

Field	Length (bits)
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
Reserved	5

MessageID	The access network shall set this field to 0x0b.
PilotID	The access network shall set this field to the PilotID for which the SectorID is requested.
ChannelBandIncluded	This field determines if a ChannelBand is included in the message.
ChannelBand	This field shall be omitted if ChannelBandIncluded is equal to 0; otherwise, the access network shall set this field to the ChannelBand record (see 15.2.1) of the sector for which the SectorID is requested.
Reserved	The access terminal shall ignore this field.

Channels	FTC
Addressing	Unicast

RLP	Best effort
AuthTag	Required if key available

11.7.5.12 SectorIDResponse

The access terminal sends a SectorIDResponse message to help the access network maintain a mapping between PilotID and SectorID values (see Table 311).

Table 311—SectorIDResponse

Field	Length (bits)
MessageID	8
SectorID	128
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
Reserved	5

MessageID	The access terminal shall set this field to 0x0a.
SectorID	The access terminal shall set this field to the SectorID of the reported sector.
PilotID	The access terminal shall set this field to the PilotID for which the SectorID is reported.

ChannelBandIncluded This field shall determine if a ChannelBand is included in the message. The access terminal shall set this field to 1 if the corresponding SectorIDRequest message had this field equal to one.

ChannelBand This field shall be omitted if ChannelBandIncluded is equal to 0; Otherwise, the access terminal shall set this field to the ChannelBand record (see 15.2.1) of the sector for which the SectorID is reported.

Reserved The access network shall ignore this field.

Channels	RTC

RLP	Best effort

Addressing	Unicast

AuthTag	Required if key available

11.7.5.13 ChannelSwitch

The access network sends a ChannelSwitch message to force the access terminal to handoff to a particular member of the Active Set (see Table 312).

Table 312—ChannelSwitch message

Field	Length (bits)
MessageID	8
PilotID	10
ChannelBandIncluded	1
ChannelBand	Variable
Reserved	5

MessageID The access terminal shall set this field to 0x0b.

PilotID This field shall be set to the PilotID of the sector to which this message is directed.

ChannelBandIncluded This field shall be set to '1' if the ChannelBand of the sector this message refers to is included in the message. The sender shall set this field to 1 if the Active Set includes more than one ChannelBand.

ChannelBand This field shall be included if ChannelBandIncluded is equal to '1', and shall be set to the ChannelBand record (see 15.2.1) of the sector to which this message refers. This field shall be omitted otherwise.

Reserved The receiver shall ignore this field.

Channels	FTC

RLP	Best effort

Addressing	Unicast

AuthTag	Required if key available

11.7.6 Interface to other protocols

11.7.6.1 Commands

This protocol issues the following command:

- *RouteControlProtocol.CreateNewStack* (AT only)

11.7.6.2 Indications

This protocol registers to receive the following indications:

- *OverheadMessages.SectorParametersUpdated*
- *ForwardLinkControlSegmentMAC.AccessGrantReceived*
- *AirLinkManagement.IdleInitiated*

11.7.7 Configuration attributes

The following complex attributes and default values are defined. The access terminal should not initiate modification of the following attributes.

11.7.7.1 Simple attributes

The configurable simple attributes are listed in Table 313.

Table 313—Configurable simple attributes

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0000	SectorIDRequestMemoryCounter	Soft Allowed/ Static	0x02	Number of previously visited sectors that the access terminal responds to for SectorIDRequest.
			0x01–0xff	Number of previously visited sectors that the access terminal responds to for SectorIDRequest.
			All other values	Reserved.
0x0001	MaxSupportedActiveSetSize Capability: This is a capability attribute and the access network shall not propose values for this attribute.	Soft Allowed/ Static	0x04	Maximum size of the combined Active Set that the access terminal supports per ChannelBand.
			0x04–0x10	Maximum size of the combined Active Set that the access terminal supports per ChannelBand.
			All other values	Reserved.
0x0002	CellNullCQIReportingEnabled Capability: This is a capability attribute and the access network shall not propose values for this attribute.	Soft Allowed/ Static	0x00	Access terminal does not support CellNullCQI reporting.
			0x01	Access terminal supports CellNullCQI reporting.
			All other values	Reserved.

Table 313—Configurable simple attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0003	DisjointLinkSupported	Soft Allowed/ Static	0x00	The access terminal does not support disjoint links.
	Capability: This is a capability attribute and the access network shall not propose values for this attribute.		0x01	The access terminal supports disjoint links.
			All other values	Reserved.
0x0004	ATChannelSwitchThreshold	Soft Allowed/ Static	0xff	This attribute does not require any restriction on inter-channel handoff.
			0x00–0xfe	This attribute, encoded as $\lfloor -2 \times 10 \times \log_{10} \text{PS} \rfloor$, where PS is the pilot strength threshold above which the access terminal is not allowed to initiate inter-channel handoff.

11.7.7.2 Complex attributes**11.7.7.2.1 SetManagementParameters attribute**

This attribute controls the addition and deletion of pilots to the Active Set.

This attribute has AttributeID 0x8000 (see Table 314).

Table 314—SetManagementParameters attribute

Field	Length (bits)	Default Value
PilotAdd	6	0x0e
PilotCompare	6	0x05
PilotDrop	6	0x12
PilotDropTimer	4	3
Padding	variable	N/A

PilotAdd This value is used by the access terminal to trigger a PilotReport in the Connected State. The access network shall set this field to the pilot detection threshold, expressed as an unsigned binary number equal to $\lfloor -2 \times 10 \times \log_{10} E_c/I_0 \rfloor$. The value used by the access terminal is -0.5 dB times the value of this field. The access terminal shall support all of the valid values specified by this field.

PilotDrop This value is used by the access terminal to start a pilot drop timer for a pilot in the Active Set or the Candidate Set. The access network shall set this field to the pilot drop threshold, expressed as an unsigned binary number equal to $\lfloor 2 \times 10 \times \log_{10} E_c/I_0 \rfloor$. The value used by the access terminal is -0.5 dB times the value of this field. The access terminal shall support all of the valid values specified by this field.

- PilotCompare

Active Set versus Candidate Set comparison threshold, expressed as a two’s complement number. The access terminal transmits a PilotReport message when the strength of a pilot in the Candidate Set exceeds that of a pilot in the Active Set by this margin. The access network shall set this field to the threshold Candidate Set pilot to Active Set pilot ratio, in units of 0.5 dB. The access terminal shall support all of the valid values specified by this field.
- PilotDropTimer

Timer value after which an action is taken by the access terminal for a pilot that is a member of the Active Set or Candidate Set, and whose strength has not become greater than the value specified by PilotDrop. If the pilot is a member of the Active Set, a PilotReport message is sent in the Connected State. If the pilot is a member of the Candidate Set, it will be moved to the Remaining Set. The access network shall set this field to the drop timer value shown in Table 315 corresponding to the pilot drop timer value to be used by access terminals. The access terminal shall support all of the valid values specified by this field.

Table 315—Pilot drop timer values

PilotDropTimer	Timer expiration (seconds)
0	< 0.1
1	1
2	2
3	4
4	6
5	9
6	13
7	19
8–15	Reserved

- Padding

The access network shall set this field to zero. The access terminal shall ignore this field. The length of this field shall be such that the entire record is octet-aligned.

Commit	Soft allowed	Scope	Dynamic

11.7.7.2.2 SupportedChannelCapability

This attribute has AttributeID 0x8001 (see Table 316).

Table 316—SupportedChannelCapability attribute

Field	Length (bits)	Default value
MaxActiveDataCarriers	8	0x01
MaxDigitalBandwidth	16	125
MaxRFBandwidth	16	125

Table 316—SupportedChannelCapability attribute (continued)

Field	Length (bits)	Default value
MaxActiveSetCarriers	8	1
Padding	variable	N/A

MaxActiveDataCarriers This field shall be set to the number of forward link channels the access terminal can demodulate simultaneously.

MaxDigitalBandwidth This field shall be set to the sum of the bandwidth of channels the access terminal can demodulate simultaneously, in units of 10 kHz.

MaxRFBandwidth This field shall be set to the maximum span of radio carriers that the access terminal can demodulate simultaneously, in units of 10 kHz.

MaxActiveSetCarriers This field shall be set to the maximum number of carriers that the access terminal can support in the active set simultaneously.

Padding The access network shall set this field to zero. The access terminal shall ignore this field. The length of this field shall be such that the entire record is octet-aligned.

Commit	Soft allowed

Scope	Dynamic

11.7.8 Protocol numeric constants

Protocol numeric constants are listed in Table 317.

Table 317—Protocol numeric constants

Constant	Meaning	Value
$N_{ASMPType}$	Type field for this protocol	See 16.1
$N_{ASMPBasic}$	Subtype field for this protocol	0x00
$N_{ASMPActive}$	Maximum size of the Active Set	8
$N_{ASMPCandidate}$	Maximum size of the Candidate Set	6

11.7.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records. The parameter records consist of the configuration attributes of this protocol.

12. Session Control Plane

12.1 Introduction

12.1.1 General overview

The Session Control Plane contains the protocol used to negotiate a session between the access terminal and the access network.

A session is a shared state maintained between the access terminal and the access network, including information such as the following:

- A unicast access terminal identifier (UATI) and Paging Identifier (PagingID) assigned to the access terminal.
- The set of protocols used by the access terminal and the access network to communicate over the air-link.
- Configuration settings for these protocols (authentication keys, parameters for Connection Layer and Lower MAC Sublayer protocols, etc.).

During a single session the access terminal and the access network can open and close a connection multiple times; therefore, sessions will be closed rarely, and only on occasions such as the access terminal leaving the coverage area or such as prolonged periods in which the access terminal is unavailable.

The Session Control Plane contains the following protocol:

- Session Control Protocol: This protocol provides the means to provision the protocols used during the session, and negotiates the configuration parameters for these protocols. In addition, this protocol ensures the session is still valid and manages closing of the session.

The protocol in the Session Control Plane is illustrated in Figure 157.

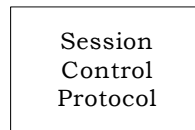


Figure 157—Session Control Plane Protocols

Since the messaging records for this protocol can be quite large, the messaging is highly optimized to minimize unnecessary overhead where possible. For example the “Total Length” of many messages depends on the receiver parsing through the message, combining fixed lengths for certain elements with one or more variable length designations to arrive at the “Total Length.” Readers should not confuse the “Length” parameter in this subclause with “Total Length” but interpret these parameters as indicative of the “element length.”

12.1.2 Data encapsulation for the InUse protocol instance

The protocol of the Session Control Plane is a control protocol and thus does not process packets.

12.2 Basic Session Control Protocol

12.2.1 Overview

The Basic Session Control Protocol provides for the negotiation and configuration of the set of protocols used during a session. This protocol also periodically ensures that the session is still valid and manages closing the session.

This protocol operates in one of eight states:

- Inactive State: This state applies only to the access terminal. In this state there are no communications between the access terminal and the access network.
- ATI Assign State: In this state the access network assigns a UATI and PagingID to the access terminal.
- Open State: In this state, a UATI and a PagingID have been assigned to the access terminal and the session is open. In this state the access terminal may initiate the full session configuration procedure or fast configuration procedure and the access network may request the access terminal to initiate the full session configuration procedure or the access network may initiate the fast configuration procedure.
- AT Initiated State: In this state, negotiation is performed at the initiative of the access terminal.
- AN Initiated State: In this state, negotiation is performed at the initiative of the access network.
- AT Fast Configuration State: In this state, fast configuration of the InUse Personality is performed at the initiative of the access terminal.
- AN Fast Configuration State: In this state, fast configuration of the InUse Personality is performed at the initiative of the access network.
- WaitingToClose State: This state applies only to the access network. In this state the access network waits for the session close procedure to complete.

Figure 158 provides an overview of the access terminal states and state transitions and Figure 159 provides an overview of the access network states and state transitions.

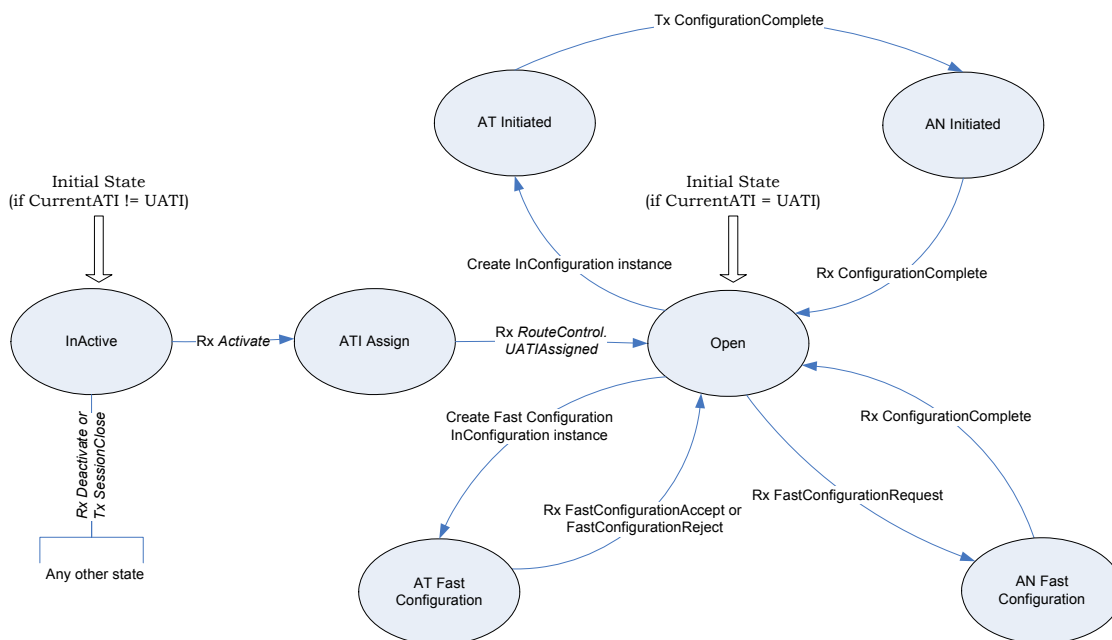


Figure 158—Basic Session Control Protocol State Diagram (access terminal)

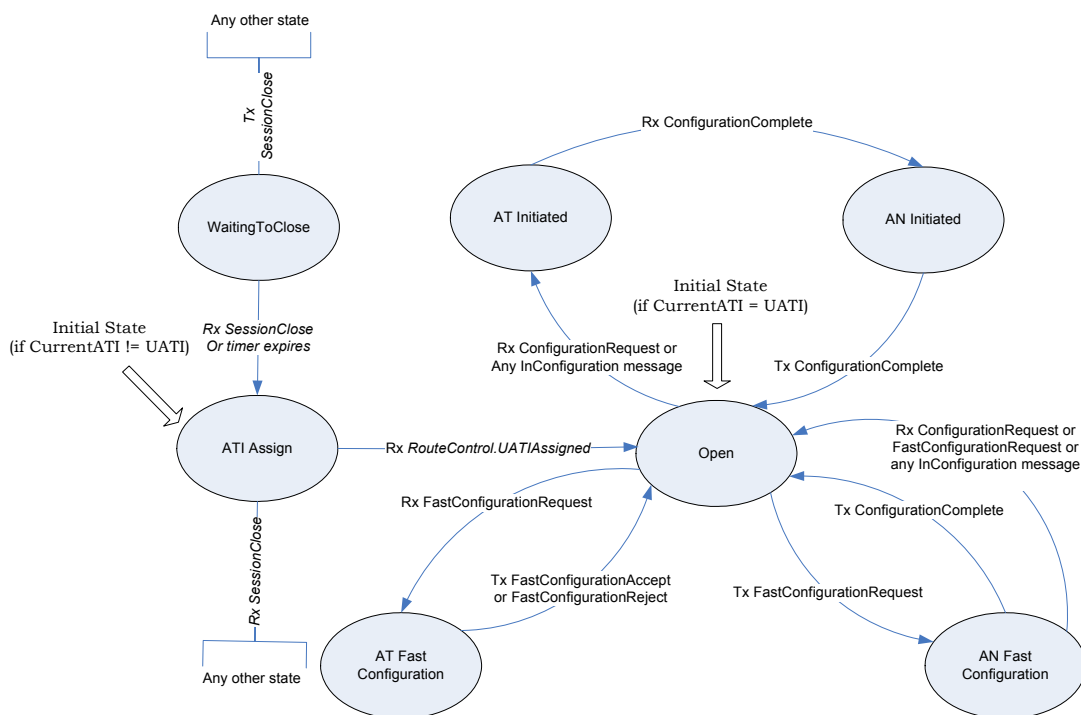


Figure 159—Basic Session Control Protocol State Diagram (access network)

For configuration of the InUse or stored personalities, this protocol supports two phases of negotiation:

- Access terminal initiated negotiation: In this phase negotiation exchanges are initiated by the access terminal. This phase is used to determine the protocols that will be used in the session and negotiate some of the protocols' parameters (e.g., Radio Link Protocol sequence lengths). Once the protocol subtypes to be instantiated are conveyed via the PersonalityIndex or ProtocolSetIdentifier field of the ConfigurationStart message or initial ConfigurationRequest message, protocol subtypes cannot be modified.
- Access network initiated negotiation: In this phase negotiation exchanges are initiated by the access network.

For fast configuration of the InUse Personality, this protocol supports two modes of negotiation:

- Access terminal initiated fast configuration: In this mode of negotiation, the access terminal proposes values for one or more attributes of the InUse Personality. Access network may accept or reject the request.
- Access network initiated fast configuration: In this mode of negotiation, the access network proposes values for one or more attributes of the InUse Personality. Access terminal may accept or reject the request.

Additional protocols may be negotiated without further modifications to the Basic Session Control Protocol.

12.2.2 Primitives, local common data, and public data

12.2.2.1 Commands

This protocol defines the following commands:

- *Activate*
- *Deactivate*

12.2.2.2 Return indications

This protocol returns the following indications:

- *Reconfigured* (to all routes)
- *SessionOpened*
- *SessionClosed* (to all routes)

12.2.2.3 Local common data

This protocol does not define any local common data.

12.2.2.4 Public data

This protocol shall make the following data public:

- Subtype for this protocol
- State of this protocol
- Busy
- PendingPersonalityIndex
- All data defined as Static Attribute, Static Non-Attribute Data, and local common data

12.2.3 Protocol Data Unit

The transmission unit of this protocol is a message. This is a control protocol; and, therefore, it does not carry payload on behalf of other layers or protocols.

This protocol uses the Signaling Protocol to transmit and receive messages.

12.2.4 Procedures and messages for the InConfiguration instance of the protocol

12.2.4.1 Procedures

This protocol uses the InUse instance of the Session Control Protocol to define the processing of the configuration messages.

12.2.4.1.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

12.2.4.1.2 Other procedures

If ATInitiatedConfigurationAllowed attribute is equal to 0x0001, the access terminal shall perform the following:

- If the HardwareID attribute is set to its default value, then the access terminal shall initiate negotiation of the HardwareID attribute.
- If the ATSupportedProtocolSetIdentifiers attribute is set to its default value, then the access terminal shall initiate negotiation of the ATSupportedProtocolSetIdentifiers attribute. If the ATSupportedApplicationProtocolIDs attribute is set to its default value, then the access terminal shall initiate negotiation of the ATSupportedApplicationProtocolIDs attribute, including all the Application Subtypes supported by the access terminal.

12.2.4.2 Message formats

This protocol does not define any messages.

12.2.5 Procedures and messages for the InUse instance of the protocol

12.2.5.1 Procedures

InConfiguration instances of this and other protocols use the messages defined by this protocol for configuration of their attributes.

The Basic Session Control Protocol allows for the negotiation and storage of up to PersonalityCount personalities. A Personality is defined as a complete set of negotiated protocol subtypes, applicationProtocolIDs, attribute values, non-attribute data values, and public data that can be used as the InUse instance of the protocols and applications. Only one Personality is in use at a time on a given Route. The total number of stored personalities shall not exceed PersonalityCount. Each Personality is identified by its PersonalityIndex. Valid values of PersonalityIndex are from zero to PersonalityCount – 1, inclusive.⁴⁰

⁴⁰The personality bin reserved for IPSI-based stack (i.e., PersonalityIndex of 0xf) is not included in PersonalityCount.

If the access terminal determines that none of the IPSIs advertised by the access network corresponds to the IPSI of any of the stored personalities, the access terminal should attempt to open a Route to this access network.

12.2.5.1.1 Generic configuration procedures

The Basic Session Control Protocol provides a means to negotiate protocol parameters.

The protocol uses ConfigurationRequest message, ConfigurationCopyRequest message, ConfigurationResponse message, ConfigurationCopyAccept message, ConfigurationCopyReject message, FastConfigurationRequest message, FastConfigurationAccept message, FastConfigurationReject message, AttributeReset message, and AttributeResetAccept message to negotiate a mutually acceptable configuration.

The initiator shall increment the TransactionID for each new ConfigurationRequest message, ConfigurationCopyRequest message, FastConfigurationRequest, or AttributeReset message sent.

The initiator uses the ConfigurationRequest message to provide the responder with a list of acceptable attribute values for each attribute. The responder uses the ConfigurationResponse message to provide the initiator with the accepted attribute value for each attribute, choosing the accepted attribute value from the initiator's acceptable attribute value list. The initiator shall order the acceptable attribute values for each attribute in descending order of preference. The initiator shall send these ordered attribute-value lists to the responder using one or more ConfigurationRequest messages. If the ordered attribute value lists fit within one ConfigurationRequest message, then the initiator should use one ConfigurationRequest message. If the ordered attribute value lists do not fit within one ConfigurationRequest message, then the initiator may use more than one ConfigurationRequest message. Each ConfigurationRequest message shall contain one or more complete ordered attribute value lists; an ordered attribute value list for an attribute shall not be split within a ConfigurationRequest message and shall not be split across multiple ConfigurationRequest messages. After sending a ConfigurationRequest message, the sender shall set the value of all attributes that were listed in the message to NULL. After receiving a ConfigurationRequest message, the responder shall respond within $T_{\text{Turnaround}}$:

- For each attribute included in the ConfigurationRequest message, the responder shall choose an acceptable attribute value from the associated acceptable attribute value list.
- If the responder does not recognize an attribute in the associated attribute list, then the responder shall skip the attribute.
- If the responder does not find an acceptable attribute value in the associated attribute list, the responder shall skip the attribute and set the attribute value to its fall-back value.

The responder shall send the accepted attribute value for each attribute within one ConfigurationResponse message. The value included for each attribute shall be one of the values listed in the ConfigurationRequest message. After receiving a ConfigurationResponse message, the initiator shall pair the received message with the associated ConfigurationRequest message using the TransactionID field of the messages. If the ConfigurationResponse message does not contain an attribute found in the associated ConfigurationRequest message, then the initiator shall set the missing attribute to its fall-back value.

The initiator uses the ConfigurationCopyRequest message to provide the responder with a proposed Personality index from where the attributes values for all the attributes in this protocol are to be taken. The subtype for this protocol shall be same as the protocol subtype in the proposed Personality index from where the configuration copy operation is requested. The responder uses the ConfigurationCopyAccept message or ConfigurationCopyReject message to accept or reject the proposed configuration copy operation, respectively. After sending a ConfigurationCopyRequest message, the sender shall set the value of all the attributes of the protocol to NULL. After receiving a ConfigurationCopyRequest message, the responder shall respond with either ConfigurationCopyAccept or ConfigurationCopyReject within $T_{\text{Turnaround}}$.

- If the responder accepts the proposed configuration copy operation, the responder shall perform the following:
 - The responder shall send the ConfigurationCopyAccept message.
 - The responder shall copy all the attribute values from the corresponding protocol in the specified Personality index.
- If the responder does not accept the proposed configuration copy operation, the responder shall perform the following:
 - The responder shall send the ConfigurationCopyReject message, and shall use the fall-back values.

After receiving a ConfigurationCopyAccept message or ConfigurationCopyReject message, the initiator shall pair the received message with the associated ConfigurationCopyRequest message using the TransactionID field of the messages.

- If the initiator receives a ConfigurationCopyAccept message, the initiator shall copy all the attribute values from the corresponding protocol in the Personality index as specified in the associated ConfigurationCopyRequest message.
- If the initiator receives a ConfigurationCopyReject message, the initiator shall set all the attributes of this protocol to their fall-back values.

The initiator uses the AttributeReset message to request the responder to set the value of the attributes included in the message to their default value. After sending a AttributeReset message, the sender shall set the value of all attributes that were listed in the message to their default values. The responder shall send an AttributeResetAccept message.

If the initiator requires no further configuration of negotiated protocols and if the value of any of the parameters for which the initiator has sent a ConfigurationRequest message or ConfigurationCopyRequest message is NULL, then the sender shall declare a failure. The initiator and the responder shall use the attribute values in the ConfigurationResponse messages and ConfigurationCopyAccept messages as the configured attribute values, provided that the attribute values were also present in the associated ConfigurationRequest message and ConfigurationCopyRequest message.

The Basic Session Control Protocol also provides a means to perform fast negotiation of protocol parameters of the InUse instance. The procedure consists of the initiator sending a single FastConfigurationRequest message, listing one or more attributes and for each attribute one or more allowed values. The initiator shall order the acceptable attribute values for each attribute in descending order of preference. If the responder finds at least one acceptable value for each listed attribute, then the responder shall send a FastConfigurationAccept message; otherwise, the responder shall send a FastConfigurationReject message. If the initiator receives a FastConfigurationReject message, the initiator shall assume that all the attributes of this protocol are using the fall-back values.

12.2.5.1.2 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- If CurrentATI public data of the Route Control Protocol is of type UATI, the protocol at the access network shall enter the Open State; otherwise, it shall enter the ATI Assign State.

If CurrentATI public data of the Route Control Protocol is of type UATI, the protocol at the access terminal shall enter the Open State; otherwise, it shall enter the Inactive State.

12.2.5.1.3 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

12.2.5.1.4 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

12.2.5.1.5 Processing the *Activate* command

If the access terminal receives the *Activate* command in the Inactive State, it shall transition to the ATI Assign State. If the access terminal receives the *Activate* command in any state other than the Inactive State, the command shall be ignored.

The access network shall ignore the command.

12.2.5.1.6 Processing the *Deactivate* command

If the access terminal receives a *Deactivate* command in the Inactive State, the command shall be ignored.

If the access terminal receives a *Deactivate* command in any state other than the Inactive State and this is the SessionAnchor Route, the access terminal shall perform the following:

- Send a SessionClose message to the access network.
- Return a *SessionClosed* indication.
- Transition to the Inactive State.

If the access network receives a *Deactivate* command in the WaitingToClose State, the command shall be ignored.

If the access network receives a *Deactivate* command in any state other than the WaitingToClose State, the access network shall send a SessionClose message and transition to the WaitingToClose State.

12.2.5.1.7 Processing the SessionClosed Indication

If the access terminal receives a *SessionControl.SessionClosed* indication from another Route while in the Inactive State, the command shall be ignored. If the access terminal receives a *SessionControl.SessionClosed* indication from another Route while in any state other than the Inactive State, the access terminal shall perform the following:

- Transition to the Inactive State.

12.2.5.1.8 Processing the SessionClose message

If the access terminal receives a SessionClose message in the Inactive State, the message shall be ignored.

If the access terminal receives a SessionClose message in any state other than the Inactive State, the access terminal shall perform the following:

- Send a SessionClose message to the access network.
- Return a *SessionClosed* indication.
- Transition to the Inactive State.

If the access network receives a *SessionClose* message in the *WaitingToClose* State, the access network shall process it as specified in 12.2.5.1.19.

If the access network receives a *SessionClose* message in any state other than the *WaitingToClose* State, the access network shall:

- Return a *SessionClosed* indication.
- Transition to the ATI Assign State.

12.2.5.1.9 Processing the reconfigured indication

If the access terminal receives a *SessionControl.Reconfigured* indication from another Route while in the *Inactive* State, the command shall be ignored. If the access terminal receives a *SessionControl.Reconfigured* indication from another Route while in any state other than the *Inactive* State, the access terminal shall perform the following:

- The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.
- The access terminal shall send a *SessionUpdated* message to the access network, conveying the updated value of session signature.

12.2.5.1.10 Keep-alive functions

If this is the *SessionAnchor* Route and *TimerBasedRegPeriod* is not set to 0x0000, the access terminal shall perform the following session keep alive procedures:

- When this Route became the *SessionAnchor* Route, the access terminal shall start the timer-based registration timer, with the value initialized to *TimerBasedRegPeriod*.
- If the access terminal receives a *AirlinkManagement.RegistrationSucceeded* indication, the access terminal shall reset the timer-based registration timer.
- If the timer-based registration timer expires, the access terminal shall send *AirlinkManagement.SendRegistration* command and shall reset the timer-based registration timer.

If this is the *SessionAnchor* Route and *TimerBasedRegPeriod* is not set to 0x0000, the access network shall perform the following session keep alive procedures:

- The access network may close the session if it does not receive any registrations from the access terminal for a period equal to *SCPCloseTime*.

12.2.5.1.11 Busy maintenance

The access terminal shall set the *Busy* public data to TRUE whenever the access terminal is in the AT initiated state, AN initiated state, AT Fast Configuration State, or AN Fast Configuration state; otherwise, the access terminal shall set the *Busy* public data to FALSE.

The access network shall set the *Busy* public data to TRUE whenever the access network is in the AT initiated state, AN initiated state, AT Fast Configuration State, or AN Fast Configuration state; otherwise, the access network shall set the *Busy* public data to FALSE.

12.2.5.1.12 Inactive State

This state only applies to the access terminal. In this state there are no communications between the access terminal and the access network. The access terminal does not maintain any session-related state and the access network may be unaware of the access terminal's existence within its coverage area when the access terminal's Session Control Protocol is in this state.

In this state the protocol waits for the *Activate* command. See 12.2.5.1.5 for processing of the *Activate* command in this state.

12.2.5.1.13 ATI Assign State

In this state the Session Control Protocol in the access terminal waits for *UATIAssigned* indication from the Route Control Protocol.

12.2.5.1.13.1 Access terminal requirements

If the access terminal receives an *RouteControl.UATIAssigned* indication, it shall perform the following:

- Return a *SessionOpened* indication.
- Transition to the Open State.

12.2.5.1.13.2 Access network requirements

If the access network receives an *RouteControl.UATIAssigned* indication, it shall perform the following:

- Return a *SessionOpened* indication.
- Transition to the Open State.

12.2.5.1.14 Open State

In the Open State the access terminal has an assigned UATI and the access terminal and the access network have a session. The access terminal and the access network shall support the keep-alive mechanism defined in 12.2.5.1.10. In this protocol state the access terminal and the access network can initiate session negotiation.

12.2.5.1.14.1 Access terminal requirements

While in this state, the access terminal may start the negotiation process by entering the AT-initiated state or AT Fast Configuration state. If the access terminal wants to change the value of any session data, the access terminal shall first enter the AT-initiated state or AT Fast Configuration state and send a *ConfigurationRequest* or *FastConfigurationRequest* message, respectively, to start the configuration process. If the access terminal wants to configure a Personality, the access terminal shall first enter the AT-initiated state and send a *ConfigurationRequest* message to start the configuration process.

While in this state, the access terminal may send a *ConfigurationStartRequest* message to request the access network to start session negotiation for creating a new Personality.

The access terminal shall not start the session negotiation process:

- If the access terminal is performing session configuration with another access network (i.e., an *InConfiguration* instance has been created).
- If the value of the *ATInitiatedConfigurationAllowed* attribute is equal to 0x0000.

To start the session negotiation process, the access terminal shall transition to the AT Initiated State or the AT Fast Configuration State.

While in this state, the access terminal may send a *SwitchPersonalityRequest* message to the access network to suggest switching the current *InUse* Personality of that access network. The access terminal shall not send this request unless the Route Control Protocol State public data of the Route Control Protocol is set to Open.

The access terminal shall ignore a ConfigurationComplete message, FastConfigurationAccept message, or DeletePersonalityRequest message received in this state.

If the access terminal receives a SwitchPersonality message, then the access terminal shall perform the following:

- If the value of the SessionSignature field in the message is not equal to the value at the access terminal, the access terminal shall send a SwitchPersonalityReject message to the access network, with the RejectReason set to 'SessionSignature out of sync' and including the value of the SessionSignature at the access terminal.
- If the HardSwitch field of the SwitchPersonality message is set to '0' and Personality conveyed via the PersonalityIndex field is not soft-switchable with respect to the Personality of the InUse instance of any of the Routes (i.e., the difference between these two personalities include hard-committable attributes), then the access terminal may accept this switch Personality request. If the access terminal does not accept the request, the access terminal shall send a SwitchPersonalityReject message to the access network, with the RejectReason set to 'switch not accepted'.
- If none of the above conditions are true, then the access terminal shall accept this switch Personality request.
- If the access terminal accepts switching to the Personality indicated by PersonalityIndex field of the SwitchPersonality message, the access terminal shall perform the following:
 - The access terminal shall send a SwitchPersonalityAccept message to the access network.
 - If the HardSwitch field of the SwitchPersonality message is set to '0', the access terminal shall perform the following:
 - The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.
 - If the HardSwitch field of the SwitchPersonality message is set to '1', the access terminal shall perform the following:
 - The access terminal shall set PendingPersonalityIndex to PersonalityIndex field of the SwitchPersonality message.
 - Issue a *AirlinkManagement.CloseConnection* command.
- If the access terminal receives a ConfigurationStart message or a FastConfigurationRequest message, the access terminal shall perform the following:
- If the value of the SessionSignature field in the message is not equal to the value at the access terminal, the access terminal shall send a ConfigurationReject message to the access network, with the RejectReason set to 'SessionSignature out of sync' and shall include the value of the SessionSignature at the access terminal.
- If the access terminal is currently performing session configuration with another access network (i.e., an InConfiguration instance has been created), the access terminal shall send a ConfigurationReject message to the access network, with the RejectReason set to 'Session Configuration in progress'.
- If none of the above conditions are true, the access terminal shall accept this request.
- If the access terminal accepts this request, the access terminal shall perform the following:
 - If the message being accepted is the ConfigurationStart message, the protocol in the access terminal shall transition to the AT Initiated State.
 - If the message being accepted is the FastConfigurationRequest message, the protocol in the access terminal shall transition to the AN Fast Configuration State.

12.2.5.1.14.2 Access network requirements

While in this state, the access network may send a ConfigurationStart message or a FastConfigurationRequest message to start the negotiation process (e.g., the access network may send this

message to negotiate a new stream). If the access network wants to change any session data, the access network shall first send a ConfigurationStart message or a FastConfigurationRequest message to start the configuration process. If the access network wants to create or delete personalities, the access network shall first send a ConfigurationStart message to start the configuration process. While in this state, the access network may send a SwitchPersonality message to request switching of its InUse Personality to another stored Personality. The access network shall not send a Personality switch request unless the Route Control Protocol State public data of the Route Control Protocol is set to Open.

If the access network sends a ConfigurationStart message, the access network shall transition to the AT Initiated State.

If the access network sends a FastConfigurationRequest message for fast configuration of the InUse Personality, the access network shall perform the following:

- For each attribute included in this message, the access network shall include one or more proposed values, in the order of decreasing preference.
- Upon sending the message, the access network shall transition to the AN Fast Configuration State.

If the access network receives a ConfigurationRequest message or a FastConfigurationRequest message, the access network shall perform the following:

- If the value of the SessionSignature field in the message is not equal to the value at the access network, the access network shall send a ConfigurationReject message to the access terminal, with the RejectReason set to 'SessionSignature out of sync' and including the value of the SessionSignature at the access network.
- If the request is not being rejected, the access network shall perform the following:
 - If the message being accepted is the ConfigurationRequest message, the protocol in the access network shall transition to the AT Initiated State.

If the message being accepted is the FastConfigurationRequest message, the protocol in the access network shall transition to the AT Fast Configuration State. If the access network desires switching its InUse Personality to another stored Personality, the access network shall send a SwitchPersonality message to the access terminal. The access network shall set the fields of the corresponding SwitchPersonality message as follows:

- The SessionSignature field shall be set to the current signature of the session corresponding to the PersonalityIndex conveyed in this message.
- The PersonalityIndex field shall be set to the index of the desired stored Personality to switch to. The PersonalityIndex field shall not be set to 0xf.
- The HardSwitch field shall be set to '0' if and only if the difference between the current InUse Personality and the desired stored Personality to switch to does not include any hard-committable attributes.

Upon sending the SwitchPersonality message, the access network shall perform the following:

- If the access network receives a SwitchPersonalityReject message, the access network shall not proceed with the Personality switch procedures.
- If the access network receives a SwitchPersonalityAccept message, the access network shall perform the following:
 - If the HardSwitch field of the SwitchPersonality message was set to '0', the access network shall perform the following:
 - The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.

If the current InUse Personality is based on an InitialProtocolSetIdentifier (i.e., corresponding stored Personality index equals 0xf), then the access network shall send a SwitchPersonality message to switch the current InUse Personality to a stored Personality in Personality index other than 0xf.⁴¹

If the access network receives a SwitchPersonalityRequest message, the access network may send a SwitchPersonality message to initiate switch of its InUse Personality.

If the access network receives a ConfigurationStartRequest message, the access network may send a ConfigurationStart message to initiate session negotiation procedures.

12.2.5.1.15 AT Initiated State

During the AT Initiated State of the Basic Session Control Protocol, session configuration is performed with the access terminal being the initiator of each exchange. The access terminal and the access network use the messages defined in this protocol to configure their associated parameters that will be used for the session.

The protocol in the access terminal or the access network shall return a *Failed* indication and transition to the Inactive state, if any of the negotiated protocols declares a failure.

12.2.5.1.15.1 Access terminal requirements

Upon entering this state, the access terminal shall perform the following:

- If the access terminal entered this state due to accepting a ConfigurationStart message from the access network, then the InConfiguration instances below are created based on the value of the PersonalityIndexOrPSI field in the ConfigurationStart message received from the access network; otherwise, the InConfiguration instances below are created based on the value of the PersonalityIndexOrPSI field in the ConfigurationRequest message sent by the access terminal.
- Create an InConfiguration protocol instance for each of the protocols as follows:
 - If the InConfiguration instances are being created from a stored Personality (i.e., PersonalityIndexOrPSI field is set to '1'), create the InConfiguration protocol instance with the same protocol ProtocolID as that of the corresponding protocol in the stored Personality.
 - If the InConfiguration instances are being created from a ProtocolSetIdentifier (i.e., PersonalityIndexOrPSI field is set to '0'), create the InConfiguration protocol instance with a ProtocolID same as the ProtocolID value of the corresponding Protocol Type of the ProtocolSetIdentifier.
- Create an InConfiguration application instances as follows:
 - If the InConfiguration instances are being created from a stored Personality (i.e., PersonalityIndexOrPSI field is set to '1'), for each stream that is bound to an application in the stored Personality, create an InConfiguration application instance with the same application ProtocolID as that of the application instance corresponding to that stream in the stored Personality.
 - If the InConfiguration instances are being created from a ProtocolSetIdentifier (i.e., PersonalityIndexOrPSI field is set to '0'), for each stream that is bound to an application as specified by the ProtocolSetIdentifier, create an InConfiguration application instance with the same application subtype as that of the application instance corresponding to that stream in the ProtocolSetIdentifier.

If the access terminal entered this state due to accepting a ConfigurationStart message from the access network, and if the access terminal sends a ConfigurationRequest message, the access terminal shall set the

⁴¹Personality index 0xf is reserved for personalities created from InitialProtocolSetIdentifier. This Personality bin needs to be vacated since the access terminal may want to add other Routes based on an InitialProtocolSetIdentifier.

PersonalityIndexOrPSI, PersonalityIndex, and ProtocolSetIdentifier fields to the corresponding values received in the ConfigurationStart message.

The access terminal shall not send a DeletePersonalityAccept message.

The access terminal shall process the configuration messages it receives (i.e., ConfigurationResponse message, ConfigurationCopyAccept message, ConfigurationCopyReject message, or AttributeResetAccept message) according to the procedures specified in 12.2.5.1.1.

The access terminal may use the protocol configuration procedures defined by the protocol to perform access terminal-initiated parameter configuration.

If the protocol in access terminal requires no further configuration of protocols, it shall send a ConfigurationComplete message to the access network and transition to the AN Initiated State.

12.2.5.1.15.2 Access network requirements

Upon entering this state, the access network shall perform the following:

- If the access network entered this state due to sending a ConfigurationStart message, then the InConfiguration instances below are created based on the value of the PersonalityIndexOrPSI field in the ConfigurationStart message. If the access network entered this state due to receiving a ConfigurationRequest message, the InConfiguration instances below are created based on the value of the PersonalityIndexOrPSI field in the ConfigurationRequest message.
- Create an InConfiguration protocol instance for each of the protocols as follows:
 - If the InConfiguration instances are being created from a stored Personality (i.e., PersonalityIndexOrPSI field is set to '1'), create the InConfiguration protocol instance with the same protocol subtype as that of the corresponding protocol in the stored Personality.
 - If the InConfiguration instances are being created from a ProtocolSetIdentifier (i.e., PersonalityIndexOrPSI field is set to '0'), create the InConfiguration protocol instance with a subtype same as the subtype value of the corresponding Protocol Type of the ProtocolSetIdentifier.
- Create an InConfiguration application instances as follows:
 - If the InConfiguration instances are being created from a stored Personality (i.e., PersonalityIndexOrPSI field is set to '1'), for each stream that is bound to an application in the stored Personality, create an InConfiguration application instance with the same application ProtocolID as that of the application instance corresponding to that stream in the stored Personality.
 - If the InConfiguration instances are being created from a ProtocolSetIdentifier (i.e., PersonalityIndexOrPSI field is set to '0'), for each stream that is bound to an application as specified by the ProtocolSetIdentifier, create an InConfiguration application instance with the same application ProtocolID as that of the application instance corresponding to that stream in the ProtocolSetIdentifier.
- Upon creating the InConfiguration protocol and application instances as specified above, if the access network receives a ConfigurationRequest message containing a different value in PersonalityIndexOrPSI, PersonalityIndex, or ProtocolSetIdentifier fields, the access network shall perform the following:
 - The access network shall purge the current InConfiguration protocol and application instances.
 - The access network shall create new InConfiguration protocol and application instances as specified above, based on the value of PersonalityIndexOrPSI, PersonalityIndex, and ProtocolSetIdentifier fields received in the ConfigurationRequest message.

If the access network receives a configuration message from the access terminal (i.e., ConfigurationRequest message, ConfigurationCopyRequest message, or AttributeReset message), it shall process it as specified in 12.2.5.1.1. If the protocol in the access network receives a ConfigurationComplete message, it shall transition to the AN Initiated State.

12.2.5.1.16 AN Initiated State

During the AN Initiated State of the protocol, the access network and the access terminal execute the access network-initiated configuration procedures specified by each negotiated protocol. These procedures typically allow the access network to override default values otherwise used by the access terminal.

The protocol in the access terminal or the access network shall return a *Failed* indication and transition to the Inactive state, if any of the negotiated protocols declares a failure.

12.2.5.1.16.1 Access terminal requirements

If the access terminal receives a configuration message from the access network (i.e., ConfigurationRequest message, ConfigurationCopyRequest message, or AttributeReset message), it shall process it as specified in 12.2.5.1.1. If the access terminal receives a DeletePersonalityRequest message, the access terminal shall perform the following:

- If all of the personalities included in the message are currently the InUse Personality of a Route, the access terminal shall send a DeletePersonalityReject message to the access network, with the RejectReason set to “Personality currently InUse;” otherwise, the access terminal shall select the first n number of available (i.e., not InUse) personalities in the list for deletion, where $n = \min(\text{NumDeletePersonality}, \text{number of available personalities for deletion})$.
- If the access terminal accepts the request, the access terminal shall send a DeletePersonalityAccept message to the access network conveying the PersonalityIndex of the stored Personalities to be deleted at the completion of this session configuration (i.e., when a ConfigurationComplete message with updated session signature is received from the access network).

The access terminal shall not send a DeletePersonalityAccept message unless as a response to a DeletePersonalityRequest message from the access network.

If the access terminal receives a ConfigurationComplete message with an updated session signature, it shall perform the following:

- The Session Control Protocol shall store the InConfiguration protocol and application instances as a Personality that is identified by the PersonalityIndexStore field of the ConfigurationComplete message.
- The access terminal shall update the static attributes and static data of all stored personalities with the negotiated values of the corresponding data.
- The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.
- The Session Control Protocol shall return a Reconfigured indication.
- The access terminal shall send a SessionUpdated message to the access network, conveying the updated value of session signature.
- The access terminal shall transition to the Open State.

12.2.5.1.16.2 Access network requirements

In this protocol state, the access network may execute the configuration procedures as specified in 12.2.5.1.1. If the access network receives a configuration messages from the access terminal (i.e.

ConfigurationResponse message, ConfigurationCopyAccept message, ConfigurationCopyReject message, or AttributeResetAccept message), it shall process it as specified in 12.2.5.1.1

If the access network desires to delete one or more stored Personalities, the access network shall send a DeletePersonalityRequest message to the access terminal conveying the PersonalityIndex of one or more stored personalities, ordered in decreasing preference for deletion, and indicating the number of personalities to be deleted. If the access network receives a DeletePersonalityAccept message, the access network shall delete the stored personalities indicated by the PersonalityIndex field of the received message at the completion of this session configuration (i.e., when a ConfigurationComplete message with updated session signature is sent to the access terminal).

The access network shall send a ConfigurationComplete message if all of the following conditions are satisfied:⁴²

- The access network does not require further changes to session data.
- The access network does not require further deletion of stored Personalities.

When sending a ConfigurationComplete message to complete the negotiation, the access network shall set the fields of the message as follows:

- SessionSignature field shall be set to one more (modulo 2^{16}) than the current value of the session signature.
- PersonalityIndexStore field shall not be set to the index of an existing Personality unless the difference between that existing Personality and the InConfiguration instance does not include any hard-committable attributes.
- PersonalityIndexStore field shall not be set to 0xf.

Upon sending a ConfigurationComplete message with updated session signature, the access network shall perform the following:

- The Session Control Protocol shall store the InConfiguration protocol and application instances as a Personality that is identified by the PersonalityIndexStore field of the ConfigurationComplete message.
- The access network shall update the static attributes and static data of all stored personalities with the negotiated values of the corresponding data.
- The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.
- The access network shall transition to the Open State.

12.2.5.1.17 AT Fast Configuration State

During the AT Fast Configuration State of the Basic Session Control Protocol, the access terminal sends a single FastConfigurationRequest message to request changes to one or more attributes of the current InUse Personality.

The protocol in the access terminal or the access network shall return a *Failed* indication and transition to the Inactive state, if any of the negotiated protocols declares a failure.

⁴²In cases where the access network wants the access terminal to send a SessionUpdated message but without negotiating the session data with the access terminal (for example when session was modified by the access network without access terminal involvement), the access network can send the ConfigurationComplete message (with an updated SessionSignature) immediately upon entering the AN-initiated State.

12.2.5.1.17.1 Access terminal requirements

Upon entering this state, the access terminal shall perform the following:

- Create an InConfiguration protocol instance for each of the protocols with the same protocol subtype as that of the corresponding protocol in the InUse Personality.
- Create an InConfiguration application instance for each of the applications with the same application ProtocolID as that of the corresponding application in the InUse Personality.

The access terminal shall send a FastConfigurationRequest message requesting changes to one or more attributes of the InUse Personality. For each attribute included in this message, the access terminal shall include one or more proposed values, in the order of decreasing preference.

If the access terminal receives a FastConfigurationAccept message, the access terminal shall perform the following:

- The Session Control Protocol shall store the InConfiguration protocol and application instances as the Personality identified by the Personality index of the current InUse Personality.
- The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.
- The access terminal shall update the static attributes and static data of all stored personalities with the negotiated values of the corresponding data.
- The Session Control Protocol shall return a *Reconfigured* indication.
- The access terminal shall send a SessionUpdated message to the access network, conveying the updated value of session signature.
- The access terminal shall transition to the Open State.

If the access terminal receives a FastConfigurationReject message, the access terminal shall perform the following:

- The access terminal shall discard all InConfiguration instances.
- The access terminal shall transition to the Open State.

12.2.5.1.17.2 Access network requirements

Upon entering this state, the access network shall perform the following:

- Create an InConfiguration protocol instance for each of the protocols with the same protocol subtype as that of the corresponding protocol in the InUse Personality.
- Create an InConfiguration application instance for each of the applications with the same application ProtocolID as that of the corresponding application in the InUse Personality.

If the access network accepts a proposed value for each of the attributes listed in the received FastConfigurationRequest message, the access network shall send a FastConfigurationAccept message; otherwise, the access network shall send a FastConfigurationReject message:

- If a FastConfigurationAccept message is being sent to the access terminal, the access network shall perform the following:
 - The access network shall include the selected value for each of the attributes listed in the corresponding FastConfigurationRequest message.
 - The access network shall set the SessionSignature field to one more (modulo 2^{16}) than the current value of the session signature.
 - Upon sending this message, the access network shall perform the following:

- The Session Control Protocol shall store the InConfiguration protocol and application instances as the Personality identified by the Personality index of the current InUse Personality.
- The access network shall update the static attributes and static data of all stored personalities with the negotiated values of the corresponding data.
- The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.
- The access network shall transition to the Open State.
- If a FastConfigurationReject message is being sent to the access terminal, the access network shall perform the following:
 - The access network shall discard the InConfiguration instances.
 - The access network shall transition to the Open State.

12.2.5.1.18 AN Fast Configuration State

During the AN Fast Configuration State of the Basic Session Control Protocol, the access network sends a single FastConfigurationRequest message to request changes to one or more attributes of the current InUse Personality.

The protocol in the access terminal or the access network shall return a *Failed* indication and transition to the Inactive state, if any of the negotiated protocols declares a failure.

12.2.5.1.18.1 Access terminal requirements

Upon entering this state, the access terminal shall perform the following:

- Create an InConfiguration protocol instance for each of the protocols with the same protocol subtype as that of the corresponding protocol in the InUse Personality.
- Create an InConfiguration application instance for each of the applications with the same application ProtocolID as that of the corresponding application in the InUse Personality.

If the access terminal accepts a proposed value for each of the attributes listed in the received FastConfigurationRequest message, the access terminal shall send a FastConfigurationAccept message; otherwise, the access terminal shall send a FastConfigurationReject message:

- If a FastConfigurationAccept message is being sent to the access network, the access terminal shall include the selected value for each of the attributes listed in the corresponding FastConfigurationRequest message.
- If a FastConfigurationReject message is being sent to the access network, the access terminal shall discard the InConfiguration instances and shall transition to the Open State.

If the access terminal receives a ConfigurationComplete message with an updated SessionSignature, the access terminal shall perform the following:

- The Session Control Protocol shall store the InConfiguration protocol and application instances as the Personality identified by the Personality index of the current InUse Personality.
- The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.
- The access terminal shall update the static attributes and static data of all stored personalities with the negotiated values of the corresponding data.
- The Session Control Protocol shall return a *Reconfigured* indication.
- The access terminal shall send a SessionUpdated message to the access network, conveying the updated value of session signature.

- The access terminal shall transition to the Open State.

12.2.5.1.18.2 Access network requirements

Upon entering this state, the access network shall perform the following:

- Create an InConfiguration protocol instance for each of the protocols with the same protocol subtype as that of the corresponding protocol in the InUse Personality.
- Create an InConfiguration application instance for each of the applications with the same application ProtocolID as that of the corresponding application in the InUse Personality.

While in this state, if the access network receives a ConfigurationRequest message or a FastConfigurationRequest message, the access network shall discard all InConfiguration instances and shall enter the Open State and shall process the received message in the Open State.

If the access network receives a FastConfigurationAccept message, the access network shall perform the following:

- The access network shall send a ConfigurationComplete message to the access terminal as follows:
 - The access network shall set the SessionSignature field to one more (modulo 2^{16}) than the current value of the session signature.
 - The access network shall set the PersonalityIndexStore field to the Personality index corresponding to the InUse Personality.
- Upon sending the ConfigurationComplete message, the access network shall perform the following:
 - The Session Control Protocol shall store the InConfiguration protocol and application instances as the Personality identified by the Personality index of the current InUse Personality.
 - The access network shall update the static attributes and static data of all stored personalities with the negotiated values of the corresponding data.
 - The Session Control Protocol shall direct all the InUse protocol instances to execute their Soft Commit procedures.
 - The access network shall transition to the Open State.

If the access network receives a FastConfigurationReject message, the access network shall perform the following:

- The access network shall discard all InConfiguration instances.
- The access network shall transition to the Open State.

12.2.5.1.19 WaitingToClose State

The WaitingToClose State is associated only with the protocol in the access network. In this state the protocol in the access network waits for a SessionClose message from the access terminal or an expiration of a timer.

The access network shall set the WaitingToClose State timer upon entering this state. The value of this timer shall be set to TimerBasedRegPeriod or $T_{SCPMinClose}$, whichever is larger.

When the access network receives a SessionClose message or when the WaitingToClose State timer expires the protocol shall:

- Return a *SessionClosed* indication.
- Transition to the ATI Assign State.

While in this state, if the access network receives any packet from the access terminal, which is addressed by the UATI assigned during this session and contains anything but a SessionClose message, it shall stay in the WaitingToClose State and perform the following:

- Discard the packet.
- Respond with a SessionClose message.

12.2.5.2 Message formats

12.2.5.2.1 SessionClose

The sender sends the SessionClose message to terminate the session (see Table 318).

Table 318—SessionClose message

Field	Length (bits)
MessageID	8
CloseReason	8
MoreInfoLen	8
MoreInfo	$8 \times \text{MoreInfoLen}$

MessageID The sender shall set this field to 0x19.

CloseReason The sender shall set this field to the close reason as shown in Table 319.

Table 319—Encoding of CloseReason Field

Field Value	Meaning	MoreInfoLen	MoreInfo
0x00	Normal Close	0	N/A
0x01	Close Reply	0	N/A
0x02	Protocol Error	0	N/A
0x03	Protocol Set Configuration Failure	0	N/A
0x04	Protocol Negotiation Error	Variable	See 12.2.5.2.1.1.
0x05	Session Signature Mismatch	0	N/A
0x06	Session Lost at Session Anchor access network	0	N/A
0x07	Session Anchor access network not reachable	0	N/A
0x08	All session resources busy	0	N/A
0x09	Authentication failure	0	N/A
All other values are reserved			

MoreInfoLen Length in octets of the MoreInfo field.

MoreInfo

Additional information pertaining to the closure. The format of this field is determined by the particular close reason.

Channels	FTC RTC	RLP	Reliable on Forward Link Best effort on Reverse Link
Addressing	Unicast	AuthTag	Required when key is available

12.2.5.2.1.1 Definition of MoreInfo record when the CloseReason is 0x04

Table 320—MoreInfo record when the CloseReason is 0x04

Field	Length (bits)
One or more occurrences of the following record:	
ProtocolType	8 or 16
SubType	8
NumAttributeRecords	8
NumAttributeRecords occurrences of the following field:	
OffendingAttributeRecord	Variable Length
Padding	Variable

ProtocolType	This field shall be set to the type of the protocol to which this attribute belongs.
SubType	The sender shall set this field to the subtype value for the protocol subtype (as defined in 16.1) for which configuration failed.
NumAttributeRecords	This field shall be set to the number of occurrences of the OffendingAttributeRecord field following this field in this record.
OffendingAttributeRecord	This field shall be set to the offending attribute record. The format of the attribute record is specified in 15.2.4.
Padding	The sender shall add Padding bits to make the length of the entire record an integer number of octets. The sender shall set these bits to '0'. The receiver shall ignore this field.

12.2.5.2.2 ConfigurationRequest

The sender sends the ConfigurationRequest message to propose new values for attributes of one or more protocols (see Table 321).

Table 321—ConfigurationRequest message

Field	Length (bits)
MessageID	8
TransactionID	8
SessionSignature	0 or 16
PersonalityIndexOrPSI	0 or 1
PersonalityIndex	0 or 4
ProtocolSetIdentifier	0 or 16
Padding	0–7 (as needed)

One or more instances of the following record

ProtocolType	8 or 16
AttributeRecord	Attribute dependent

MessageID The access terminal shall set this field to 0x00; the access network shall set this field to 0x01.

TransactionID The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.

SessionSignature If the MessageID field is set to 0x01, this field shall be omitted; otherwise, this field shall be set to the signature of the current session.

PersonalityIndexOrPSI If the MessageID field is set to 0x01, this field shall be omitted; otherwise, this field shall be set as follows:

- If this message is being sent due to an AT-initiated session negotiation: the access terminal shall set this field to '1'.
- If this message is being sent due to an AN-initiated session negotiation (i.e., a response to a ConfigurationStart message): the access terminal shall set this field to the value of the PersonalityIndexOrPSI field in the received ConfigurationStart message.

PersonalityIndex If the MessageID field is set to 0x01, this field shall be omitted; otherwise, this field shall be set as follows:

The access terminal shall omit this field if the PersonalityIndexOrPSI field is set to '0'; otherwise, the access terminal shall set this field as follows:

- If this message is being sent due to an AT-initiated session negotiation: the access terminal shall set this field to the index of the stored Personality corresponding to the current InUse Personality.
- If this message is being sent due to an AN-initiated session negotiation (i.e., a response to a ConfigurationStart message): the access terminal shall set this field to the value of the PersonalityIndex field in the received ConfigurationStart message.

ProtocolSetIdentifier If the MessageID field is set to 0x01, this field shall be omitted; otherwise, this field shall be set as follows:

The access terminal shall omit this field if the PersonalityIndexOrPSI field is set to '1'; otherwise, the access terminal shall set this field as follows: the access terminal shall set this field to the value of the ProtocolSetIdentifier field in the received ConfigurationStart message.

Padding The sender shall add Padding bits to make the length of the entire message equal to an integer number of octets. The sender shall set these bits to '0'. The receiver shall ignore this field.

ProtocolType The sender shall set this field to the protocol to which this attribute belongs.

AttributeRecord The format of this record is specified in 15.2.4.

Channels	FTC	RTC	RLP	Reliable
Addressing		Unicast	AuthTag	Required when key is available

12.2.5.2.3 ConfigurationResponse

The sender sends the ConfigurationResponse message to respond to a ConfigurationRequest message (see Table 322).

Table 322—ConfigurationResponse message

Field	Length (bits)
MessageID	8
TransactionID	8
Zero or more instances of the following record	
ProtocolType	8 or 16
AttributeRecordAck	Attribute dependent
Padding	0–7 (as needed)

MessageID The sender shall set this field to 0x02.

TransactionID The sender shall set this value to the TransactionID field of the corresponding ConfigurationRequest message.

ProtocolType The sender shall set this field to the type of the protocol to which this attribute belongs.

AttributeRecordAck An AttributeRecordAck is either simple or complex. A simple AttributeRecordAck is a copy of one simple attribute from the original configuration request. A complex AttributeRecordAck is a complex attribute copied from the original configuration request, however, the AttributeValue is stripped (given a length of zero.) Because the AttributeValue is stripped, the ValueID within the complex attribute is used to identify the attribute that has been accepted in the matching configuration request.

Padding This field shall be set to Padding bits to make the length of the entire message equal to an integer number of octets. These bits shall be set to '0'. The receiver shall ignore this field.

Channels	FTC	RTC

Addressing	Unicast

RLP	Reliable

AuthTag	Required when key is available

12.2.5.2.4 ConfigurationCopyRequest

The sender sends the ConfigurationCopyRequest message to request setting the attributes to corresponding values in the specified stored Personality (see Table 323).

Table 323—ConfigurationCopyRequest

Field	Length (bits)
MessageID	8
TransactionID	8
ProtocolType	8 or 16
PersonalityIndex	4
Padding	0–7 (as needed)

MessageID The sender shall set this field to 0x03.

TransactionID The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.

ProtocolType The sender shall set this field to the type of the protocol from which the attribute values are to be copied.

PersonalityIndex The sender shall set this field to the index of the stored Personality from which the attribute values are to be set.

Padding

This field shall be set to Padding bits to make the length of the entire message equal to an integer number of octets. These bits shall be set to '0'. The receiver shall ignore this field.

Channels	FTC	RTC	RLP	Reliable
Addressing	Unicast		AuthTag	Required when key is available

12.2.5.2.5 ConfigurationCopyAccept

The sender sends the ConfigurationCopyAccept message to accept a ConfigurationCopyRequest message (see Table 324).

Table 324—ConfigurationCopyAccept message

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID

The sender shall set this field to 0x04.

TransactionID

The sender shall set this field to the value of TransactionID field in the ConfigurationCopyRequest message to which the sender is responding.

Channels	FTC	RTC	RLP	Reliable
Addressing	Unicast		AuthTag	Required when key is available

12.2.5.2.6 ConfigurationCopyReject

The sender sends the ConfigurationCopyReject message to reject a ConfigurationCopyRequest message (see Table 325).

Table 325—ConfigurationCopyReject message

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID

The sender shall set this field to 0x05.

TransactionID

The sender shall set this field to the value of TransactionID field in the ConfigurationCopyRequest message to which the sender is responding.

Channels	FTC	RTC	RLP	Reliable
Addressing	Unicast		AuthTag	Required when key is available

12.2.5.2.7 AttributeReset

The sender sends the AttributeReset message to request setting the included attributes to their default values (see Table 326).

Table 326—AttributeReset message

Field	Length (bits)
MessageID	8
TransactionID	8
One or more instances of the following record	
ProtocolType	8 or 16
AttributeID	16

MessageID The sender shall set this field to 0x06.

TransactionID The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.

ProtocolType If length of this field is 8 bits, then the sender shall set the most significant bit of this field to '0'; otherwise, the sender shall set the most significant bit of this field to '1'. The sender shall set this field to the type of the protocol to which this attribute belongs.

AttributeID The sender shall set this field to the AttributeID of the attribute to be reset to its default value.

Channels	FTC	RTC
Addressing	Unicast	

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.8 AttributeResetAccept

The sender sends the AttributeResetAccept message to accept the request received in the AttributeReset message (see Table 327).

Table 327—AttributeResetAccept message

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The sender shall set this field to 0x07.

TransactionID

The sender shall set this field to the value of TransactionID in the AttributeReset message to which the sender is responding.

Channels	FTC RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

12.2.5.2.9 FastConfigurationRequest

The sender sends the FastConfigurationRequest message to propose new values for attributes of one or more protocols of the current InUse Personality (see Table 328).

Table 328—FastConfigurationRequest

Field	Length (bits)
MessageID	8
TransactionID	8
SessionSignature	16
One or more instances of the following record	
ProtocolType	8 or 16
AttributeRecord	Attribute dependent

MessageID

The sender shall set this field to 0x08.

TransactionID

The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.

SessionSignature

The sender shall set this field to the signature of the current session.

ProtocolType

If length of this field is 8 bits, then the sender shall set the most significant bit of this field to '0'; otherwise, the sender shall set the most significant bit of this field to '1'. The sender shall set this field to the type of the protocol to which this attribute belongs.

AttributeRecord

The format of this record is specified in 15.2.4.

Channels	FTC RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

12.2.5.2.10 FastConfigurationAccept

The sender sends the FastConfigurationAccept message to accept a FastConfigurationRequest message (see Table 329).

Table 329—FastConfigurationAccept message

Field	Length (bits)
MessageID	8
TransactionID	8
SessionSignature	0 or 16
Zero or more instances of the following record	
ProtocolType	8 or 16
AttributeRecordAck	Attribute dependent

MessageID	The access terminal shall set this field to 0x09. The access network shall set this field to 0x0a.
TransactionID	The sender shall set this value to the TransactionID field of the corresponding FastConfigurationRequest message.
SessionSignature	If the MessageID field is set to 0x09, this field shall be omitted; otherwise, this field shall be set to the signature of the updated session.
ProtocolType	If length of this field is 8 bits, then the sender shall set the most significant bit of this field to '0'; otherwise, the sender shall set the most significant bit of this field to '1'. The sender shall set this field to the type of the protocol to which this attribute belongs.
AttributeRecordAck	An AttributeRecordAck is either simple or complex. A simple AttributeRecordAck is a copy of one simple attribute from the original configuration request. A complex AttributeRecordAck is a complex attribute copied from the original configuration request, however, the AttributeValue is stripped (given a length of zero.) Because the AttributeValue is stripped, the ValueID within the complex attribute is used to identify the attribute that has been accepted in the matching configuration request.

Channels	FTC RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.11 **FastConfigurationReject**

The sender sends the FastConfigurationReject message to reject a FastConfigurationRequest message (see Table 330).

Table 330—FastConfigurationReject Message

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The sender shall set this field to 0x0b.

TransactionID The sender shall set this value to the TransactionID field of the corresponding FastConfigurationRequest message.

Channels	FTC RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

12.2.5.2.12 **ConfigurationComplete**

The sender sends the ConfigurationComplete message to indicate that it has completed the negotiation procedures performed at its initiative (see Table 331).

Table 331—ConfigurationComplete message

Field	Length (bits)
MessageID	8
TransactionID	8
SessionSignature	0 or 16
PersonalityIndexStore	0 or 4
Padding	0-7 (as needed)

MessageID The access terminal shall set this field to 0x0c. The access network shall set this field to 0x0d.

TransactionID The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.

SessionSignature If the MessageID field is set to 0x0c, this field shall be omitted; otherwise, this field shall be set to the signature of the updated session.

PersonalityIndexStore If the MessageID field is set to 0x0c, this field shall be omitted; otherwise, this field shall be set to the index of the Personality in which the InConfiguration instances of the negotiated protocols and applications are to be stored, in the range from 0 to PersonalityCount – 1, inclusive

Padding The sender shall add Padding bits to make the length of the entire message equal to an integer number of octets. The sender shall set these bits to '0'. The receiver shall ignore this field.

Channels	FTC	RTC
Addressing	Unicast	

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.13 ConfigurationStart

The access network sends this message to start a session configuration process (see Table 332).

Table 332—ConfigurationStart message

Field	Length (bits)
MessageID	8
TransactionID	8
SessionSignature	16
PersonalityIndexOrPSI	1
PersonalityIndex	0 or 4
ProtocolSetIdentifier	0 or 16
Padding	0 – 7 (as needed)

MessageID The sender shall set this field to 0x0e.

TransactionID The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.

SessionSignature The access network shall set this field to the signature of the current session.

PersonalityIndexOrPSI The access network shall set this field to '1' if the starting configuration for the Personality to be negotiated is a stored Personality; the access network shall set this field to '0' if the starting configuration for the Personality to be negotiated is an ProtocolSetIdentifier.

PersonalityIndex The access network shall omit this field if PersonalityIndexOrPSI is set to '0'; otherwise, the access network shall set this field to the index of the stored Personality that is to be the starting configuration for the Personality to be negotiated.

ProtocolSetIdentifier	The access network shall omit this field if PersonalityIndexOrPSI is set to '1'; otherwise, the access network shall set this field to the ProtocolSetIdentifier that is to be the starting configuration for the Personality to be negotiated.
Padding	The access network shall add Padding bits to make the length of the entire message equal to an integer number of octets. The access network shall set these bits to '0'. The access terminal shall ignore this field.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

12.2.5.2.14 ConfigurationReject

The access terminal sends the ConfigurationReject message to reject the ConfigurationStart message or FastConfigurationRequest message from the access network. The access network sends the ConfigurationReject message to reject the ConfigurationRequest message or FastConfigurationRequest message from the access terminal (see Table 333).

Table 333—ConfigurationReject message

Field	Length (bits)
MessageID	8
TransactionID	8
RejectReason	8
SessionSignature	16 ^a

^aThis field may be absent (see field description for conditions).

MessageID	The sender shall set this field to 0x0f.
TransactionID	The access terminal shall set this field to the value of TransactionID included in the ConfigurationStart message or FastConfigurationRequest message received from the access network to which this message is the response. The access network shall set this field to the value of TransactionID included in the ConfigurationRequest message or FastConfigurationRequest message received from the access terminal to which this message is the response.
RejectReason	The access terminal shall set this field to indicate the reason the ConfigurationStart message or FastConfigurationRequest message is being rejected, as specified in Table 334. The access network shall set this field to indicate the reason the ConfigurationRequest message or FastConfigurationRequest message is being rejected, as specified in Table 335.

Table 334—RejectReason Values for access terminal

RejectReason	Reason
00000000	SessionSignature out of sync

Table 334—RejectReason Values for access terminal (continued)

RejectReason	Reason
00000001	Session Configuration in progress
All other values	Reserved

Table 335—RejectReason Values for access network

RejectReason	Reason
00000000	SessionSignature out of sync
All other values	Reserved

SessionSignature

If the RejectReason field is not set to '00000000', the sender shall omit this field; otherwise, the sender shall include this field and set it to the signature of the current session.

Channels	FTC RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.15 DeletePersonalityRequest

The access network sends a DeletePersonalityRequest message to request deletion of one or more stored Personalities (see Table 336).

Table 336—DeletePersonalityRequest message

Field	Length (bits)
MessageID	8
TransactionID	8
NumDeletePersonality	4
PersonalityCount	4
PersonalityCount instances of the following record	
PersonalityIndex	4
Padding	0–7 (as needed)

MessageID

The access network shall set this field to 0x10.

TransactionID

The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common

TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.

NumDeletePersonality	The access network shall set this field to the number of personalities that the access terminal is to delete. The access network shall set this field to a value less than or equal to the value of the PersonalityCount field.
PersonalityCount	The access network shall set this field to the number of PersonalityIndex fields in this message.
PersonalityIndex	The access network shall set this field to the Personality that the access network is requesting to delete. The access network shall order the list in decreasing preference for deletion.
Padding	This field shall be set to Padding bits to make the length of the entire message equal to an integer number of octets. These bits shall be set to '0'. The receiver shall ignore this field.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required when key is available

12.2.5.2.16 DeletePersonalityAccept

The access terminal sends a DeletePersonalityAccept message to indicate the personalities the AN is to delete (see Table 337).

Table 337—DeletePersonalityAccept message

Field	Length (bits)
MessageID	8
TransactionID	8
PersonalityCount	4
Reserved	4
PersonalityCount instances of the following record	
PersonalityIndex	4
Padding	0–7 (as needed)

MessageID	The access terminal shall set this field to 0x11.
TransactionID	The access terminal shall set this field to the value of the corresponding field in the DeletePersonalityRequest message being accepted.
PersonalityCount	The access terminal shall set this field to the number of PersonalityIndex fields in this message.
Reserved	The receiver shall ignore this field.

PersonalityIndex The access terminal shall set this field to the Personality being deleted.

Padding This field shall be set to Padding bits to make the length of the entire message equal to an integer number of octets. These bits shall be set to '0'. The receiver shall ignore this field.

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.17 DeletePersonalityReject

The access terminal sends a DeletePersonalityReject message to reject the access network request for Personality deletion (see Table 338).

Table 338—DeletePersonalityReject message

Field	Length (bits)
MessageID	8
TransactionID	8
RejectReason	8

MessageID The access terminal shall set this field to 0x12.

TransactionID The access terminal shall set this field to the value of the corresponding field in the DeletePersonalityRequest message being rejected.

RejectReason The access terminal shall set this field to indicate the reason the DeletePersonalityRequest message is being rejected, as specified in Table 339.

Table 339—RejectReason values

RejectReason	Reason
00000000	Personality currently InUse
All other values	Reserved

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.18 SwitchPersonality

The access network sends a SwitchPersonality message to request the access terminal to switch to the indicated Personality (see Table 340).

Table 340—SwitchPersonality message

Field	Length (bits)
MessageID	8
TransactionID	8
SessionSignature	16
PersonalityIndex	4
HardSwitch	1
Reserved	3

MessageID	The access network shall set this field to 0x13.
TransactionID	The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.
SessionSignature	The access network shall set this field to the signature of the current session.
PersonalityIndex	The access network shall set this field to the Personality to which the access terminal is to switch. The access network shall not set this field to 0xf.
HardSwitch	The access network shall set this field to '1' if the Route needs to be closed for the Personality switch to take effect; otherwise, the access network shall set this field to '0'.
Reserved	This field shall be ignored by the receiver.

Channels	FTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.19 SwitchPersonalityAccept

The access terminal sends the SwitchPersonalityAccept message to indicate it has accepted the Personality switch request from the access network (see Table 341).

Table 341—SwitchPersonalityAccept message

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID The access terminal shall set this field to 0x14.

TransactionID The access terminal shall set this field to the value received in the corresponding SwitchPersonality message.

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.20 SwitchPersonalityReject

The access terminal sends the SwitchPersonalityReject message to indicate it cannot accept the Personality switch request from the access network (see Table 342).

Table 342—SwitchPersonalityReject message

Field	Length (bits)
MessageID	8
TransactionID	8
RejectReason	8
SessionSignature	0 or 16

MessageID The access terminal shall set this field to 0x15.

TransactionID The access terminal shall set this field to the value received in the corresponding SwitchPersonality message.

RejectReason The access terminal shall set this field to indicate the reason the SwitchPersonality message is being rejected, as specified in Table 343.

Table 343—RejectReason values

RejectReason	Reason
00000000	SessionSignature out of sync
00000001	Proposed Personality not accepted

Table 343—RejectReason values (continued)

RejectReason	Reason
All other values	Reserved

SessionSignature

If the RejectReason field is not set to '00000000', the access terminal shall omit this field; otherwise, the access terminal shall include this field and set it to the signature of the current session.

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.21 SwitchPersonalityRequest

The access terminal sends the SwitchPersonalityRequest message to suggest access network to switch to the indicated Personality (see Table 344).

Table 344—SwitchPersonalityRequest Message

Field	Length (bits)
MessageID	8
PersonalityIndex	4
Reserved	4

MessageID

The access terminal shall set this field to 0x16.

PersonalityIndex

The access terminal shall set this field to the suggested Personality to switch to.

Reserved

This field shall be ignored by the receiver.

Channels	FTC RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.22 SessionUpdated

The access terminal sends the SessionUpdated message to indicate to the access network the session has been updated (see Table 345).

Table 345—SessionUpdated message

Field	Length (bits)
MessageID	8
SessionSignature	16

MessageID The access terminal shall set this field to 0x17.

SessionSignature The access terminal shall set this field to the session signature corresponding to the updated session.

Channels	RTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required when key is available

12.2.5.2.23 ConfigurationStartRequest

The access terminal sends this message to request access network to start the indicated session configuration process (see Table 346).

Table 346—ConfigurationStartRequest message

Field	Length (bits)
MessageID	8
PersonalityIndexOrPSI	1
PersonalityIndex	4 ^a
ProtocolSetIdentifier	16 ^a
Padding	0–7 (as needed)

^aThis field may be absent (see field description for conditions).

MessageID The access terminal shall set this field to 0x18.

PersonalityIndexOrPSI The access terminal shall set this field to '1' if the requested starting configuration for the Personality to be negotiated is a stored Personality; the access terminal shall set this field to '0' if the requested starting configuration for the Personality to be negotiated is an ProtocolSetIdentifier.

PersonalityIndex The access terminal shall omit this field if PersonalityIndexOrPSI is set to '0'; otherwise, the access terminal shall set this field to the index of the stored Personality that is to be the starting configuration for the Personality to be negotiated.

ProtocolSetIdentifier The access terminal shall omit this field if PersonalityIndexOrPSI is set to '1'; otherwise, the access terminal shall set this field to the ProtocolSetIdentifier that is to be the starting configuration for the Personality to be negotiated.

Padding The access terminal shall add Padding bits to make the length of the entire message equal to an integer number of octets. The access terminal shall set these bits to '0'. The access network shall ignore this field.

Channels	RTC
Addressing	Unicast

RLP	Best effort
AuthTag	Required when key is available

12.2.6 Interface to other protocols

12.2.6.1 Commands

This protocol issues the following command:

- *RouteControl.CloseRoute*

12.2.6.2 Indications

This protocol registers to receive the following indication:

- *SessionControl.Reconfigured (from other routes)*
- *SessionControl.SessionClosed (from other routes)*
- *RouteControl.RouteClosed*
- *RouteControl.UATIAssigned*

12.2.7 Configuration attributes

12.2.7.1 Simple attributes

The following attribute-value pairs are defined (see 15.2.4 for attribute record format).

The simple configuration attributes are listed in Table 347. The Attribute ID fields for all these attributes are two octets in length and the value fields for these attributes are two octets in length. The access terminal shall use the values in Table 347 typed in ***bold italics*** as defaults.

Table 347—Simple Configuration attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	PersonalityCount	Hard/Static	<i>0x0004</i>	Maximum of four personalities are supported.
			0x0005 – 0x0010	Maximum number of personalities supported.
			All other values	Reserved.
0x0001	SCPCloseTime	Soft/Static	<i>0x0CA8</i>	Default is 54 hours.
			0x0000 to 0xffff	Session keep alive period in minutes.
0x0002	ATInitiatedConfigurationAllowed	Soft/Static	0x0000	Access terminal is not allowed to initiate session configuration.
			0x0001	The access terminal is allowed to initiate session configuration.
			All other values	Reserved.

Table 347—Simple Configuration attributes (continued)

Attribute ID	Attribute	Commit/ Scope	Values	Meaning
0x0003	TimerBasedRegPeriod	Soft/Static	0x003c	One hour.
			0x000e–0xffff	Timer-based registration period in minutes. This value of this attribute shall be a multiple of the PagePeriod3 public data of the Idle State Protocol.
			0x0000	Timer-based registration is disabled.
			All other values	Reserved.

12.2.7.2 Complex attributes

The following static and dynamic complex attribute and default values are defined (see 15.2.4 for attribute record definition).

12.2.7.2.1 ATSupportedProtocolSetIdentifiers attribute (access terminal capability)

The sender shall set AttributeID field to 0x8000 (see Table 348).

Table 348—ATSupportedProtocolSetIdentifiers attribute

Field	Length (bits)	Default
NumProtocolSetIdentifiers	8	1
NumProtocolSetIdentifiers occurrences of the following field:		
ProtocolSetIdentifiers	16	N/A

NumProtocolSetIdentifiers

The sender shall set this field to the number of ProtocolSetIdentifiers included.

ProtocolSetIdentifiers

This field shall be set to the Protocol Set Identifier supported by the access terminal.

Commit	Hard

Scope	Static

12.2.7.2.2 ATSupportedApplicationProtocolIDs attribute (access terminal capability)

This is a static complex attribute.

The sender shall set AttributeID field to 0x8001(see Table 349).

Table 349—ATSupportedApplicationProtocolIDs attribute

Field	Length (bits)	Default
NumAppProtocolIDs	8	0

NumAppProtocolIDs occurrences of the following field:

Table 349—ATSupportedApplicationProtocolIDs attribute (continued)

Field	Length (bits)	Default
ApplicationProtocolID	8	N/A
SupportedProtocolParametersValuesLength	8	0x00
SupportedProtocolParametersValues	SupportedProtocolParametersValuesLength × 8	N/A

NumAppProtocolIDs This field shall be set to the number of occurrences of the ApplicationProtocolID field following this field in this record.

ApplicationProtocolID This field shall be set to the ApplicationProtocolID that the access terminal supports.

SupportedProtocolParametersValuesLength
If the SupportedProtocolParametersValues record is not included, the sender shall set this field to 0x00. Otherwise, the sender shall set this field to the length of the SupportedProtocolParametersValues record in units of octets.

SupportedProtocolParametersValues
Unless specified otherwise by 16.1, the sender shall omit this record. If ApplicationProtocolID is 0x04, then the sender shall set this record as defined in 12.2.7.2.2.1. If ApplicationProtocolID is 0x08, then the sender shall set this record as defined in 12.2.7.2.2.2. If ApplicationProtocolID is less than 0x08 and not equal to 0x04, the sender shall omit this record.

12.2.7.2.2.1 Definition of SupportedProtocolParametersValues record when the ApplicationProtocolID is ROHC (ApplicationProtocolID = 0x04)**Table 350—SupportedProtocolParametersValues (ApplicationProtocolID = 0x04)**

Field	Length (bits)	Default
MaxSupportedMaxCID	16	0
LargeCIDSupported	1	0
MaxSupportedMRRU	16	0
TimerBasedCompressionSupported	1	1
SupportedProfileCount	8	4
SupportedProfileCount occurrences of the following field:		
SupportedProfile	16	0x0000 (uncompressed), 0x0001 (RTP), 0x0002 (UDP), 0x0003 (ESP)
Reserved	6	N/A

MaxSupportedMaxCID The sender shall set this field to the maximum MAX_CID parameter supported (see IETF RFC 3095).

LargeCIDSupported The sender shall set this field to '0' if large CID representation is not supported according to (see IETF RFC 3095). Otherwise, the sender shall set this field to '1' if large CID representation is supported.

MaxSupportedMRRU The sender shall set this field to the MRRU supported by the decompressor according to (see IETF RFC 3095). Default value is 0x0000 (no segmentation).

TimerBasedCompressionSupported
The sender shall set this field to '1' if the compressor at the access terminal supports timer-based compression mode. Otherwise, the sender shall set this field to '0'.

SupportedProfileCount The sender shall set this field to the number of ROHC profiles supported.⁴³

SupportedProfile The sender shall set this field to the ROHC profile supported by the compressor and decompressor.

Reserved The receiver shall ignore this field.

⁴³IANA ROHC Profile Identifiers can be found at <http://www.iana.org/assignments/rohc-pro-ids>.

12.2.7.2.2.2 Definition of SupportedProtocolParametersValues record when the ApplicationProtocolID is EAP (ApplicationProtocolID = 0x08)

Table 351—SupportedProtocolParametersValues record (ApplicationProtocolID = 0x08)

Field	Length (bits)	Default
EAPFastReauthenticationSupported	1	'0'
Reserved	7	N/A

EAPFastReauthenticationSupported

If EAP Fast Reauthentication is supported, then the sender shall set this field to '1'; otherwise, the sender shall set this field to '0'.

Reserved

The receiver shall ignore this field.

Commit	Hard

Scope	Static

12.2.7.2.2.3 HardwareID attribute

The sender shall set AttributeID field to 0x8002 (see Table 352).

Table 352—HardwareID attribute

Field	Length (bits)	Default
HardwareIDType	4	0xf
Reserved	4	N/A
HardwareIDLength	8	0x00
HardwareIDValue	8 HardwareIDLength	N/A

HardwareIDType

The access terminal shall set this field according to Table 353.

Table 353—HardwareIDType encoding

HardwareIDType field value	Meaning
0x0	HardwareID not provisioned
0x1	Mobile Equipment Identifier (MEID);
0xf	Null
Other values	Reserved

HardwareIDLength

If HardwareIDType field is not set to 0x0 or 0xf, the access terminal shall set this field to the length in octets of the HardwareIDValue field; otherwise the access terminal shall set this field to 0x00.

Reserved

The receiver shall ignore the values in this field.

HardwareIDValue The access terminal shall set this field to the unique ID (specified by **HardwareIDType**) that has been assigned to the terminal by the manufacturer.

Commit	Hard	Scope	Static

12.2.7.3 Non-attribute data

The static non-attribute data for this protocol are defined below.

12.2.7.3.1 SessionSignature

The sender shall set **DataID** field to 0x0000 (see Table 354).

Table 354—SessionSignature

Field	Length (bits)
SessionSignature	16

SessionSignature This field shall be set to the signature of the current session.

12.2.8 Protocol numeric constants

Protocol numeric constants are listed in Table 355.

Table 355—Protocol numeric constant

Constant	Meaning	Value
N _{SCPT} Type	Type field for this protocol	See 16.1
N _{SCP} Basic	ProtocolID field for this protocol	0x00
T _{SCP} MinClose	Minimum recommended timer setting for WaitingToOpen State	300 seconds
T _{Turnaround}	The elapsed duration before which a receiver shall respond to a ConfigurationRequest, FastConfigurationRequest, or ConfigurationCopyRequest	2 seconds

12.2.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records. All configuration attributes are Session State Information records. The following Non-Attribute Static Data are Session State Information record: SessionSignature. This protocol defines the following parameter records.

12.2.9.1 ProtocolSetIdentifier

The ParameterType for this record shall be set to 0x02 (see Table 356).

Table 356—ProtocolSetIdentifier

Field	Length (bits)
ProtocolSetIdentifier	16

ProtocolSetIdentifier This field shall be set to the ProtocolSetIdentifier corresponding to the Personality.

13. Route Control Plane

13.1 Introduction

Route Control Plane consists of the following protocol:

- Route Control Protocol

13.2 Basic Route Control Protocol

13.2.1 Overview

The Route Control Protocol resides in the Control Plane. The Route Control Protocol performs the following functions:

- Controls the set up and tear down of Routes.
- Maintains mapping between short Route identifier (RouteID) and long Route identifier (ANID).
- Maintains identity of the DataAttachmentPoint Route and SessionAnchor Route.
- Maintains the identity of the Forward Link Serving Route and Reverse Link Serving Route
- UATI and PagingID assignment.

A Route consists of an InUse protocol stack associated with an access network.

This protocol can be in one of four states:

- InActive State: In this state, a Route to the access network does not exist.
- WaitingToOpen State: In this state, a Route to the access network exists but is not open yet. In this state, the access terminal may send a RouteOpenRequest message to the access network and waits for the response. In this state, the access terminal has limited communication with the access network. A Route whose Route Control Protocol is in this state may carry packets bound for another Route.
- Open State: In this state, a Route to the access network exists (i.e., access network has access terminal's session) and the Route is open. In this state, the access terminal and the access network can send data and messages to each other.
- WaitingToClose State: This state applies only to the access network. In this state the access network waits for the Route close procedure to complete.

Figure 160 provides an overview of the access terminal states and state transitions.

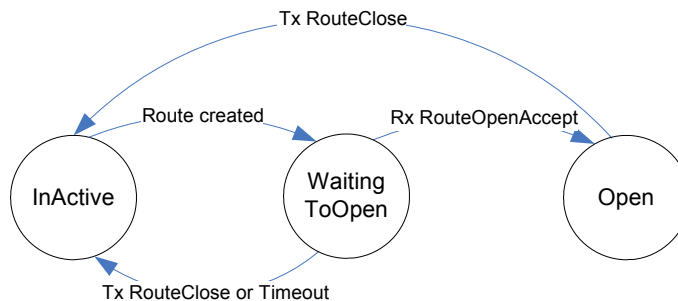


Figure 160—Route Control Protocol State Diagram (access terminal)

Figure 161 provides an overview of the access network states and state transitions.

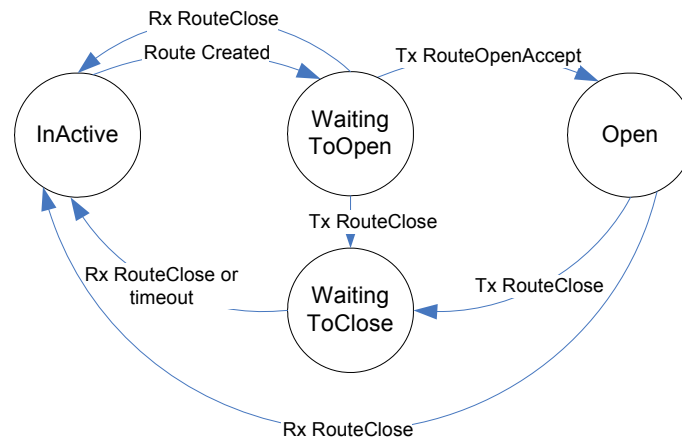


Figure 161—Route Control Protocol State Diagram (access network)

13.2.2 Primitives, local common data, and public data

13.2.2.1 Commands

This protocol defines the following commands:

- *CreateNewStack(PilotID)*
- *CloseRoute*
- *DeleteOldRoute*
- *QueryRouteMap*

13.2.2.2 Return indications

This protocol returns the following indications:

- *RouteOpened*
- *RouteClosed*
- *RouteOpenRejected*
- *UATIAssigned(RouteID)*
- *RouteCreated(RouteCounter)*
- *DAPAssigned(RouteID)*
- *LinkInitialized*

13.2.2.3 Procedure calls

- CREATE_ID_TAG
 - Inputs: SequenceNumber
 - Outputs: IDTagSize, IDTag

13.2.2.4 Local common data

This protocol defines the following local common data (i.e., AT only):

- PagingIDList
- RouteCounter
- RouteMap (RouteID and ANID of all Routes in Open State)
- DAPRefreshTime
- RetryDelay[ANID]
- CurrentATI (see 15.1)
- SessionSignature

13.2.2.5 Public data

This protocol shall make the following data public:

- Subtype for this protocol
- State of this protocol
- LinkID
- Whether this protocol belongs to the DataAttachmentPoint Route
- Whether this protocol belongs to the SessionAnchor Route
- Whether this protocol belongs to the Forward Link Serving Route
- Whether this protocol belongs to the Reverse Link Serving Route
- All data defined as Static Attribute, Static Non-Attribute Data, and local common data

13.2.3 Protocol Data Unit

The transmission unit of this protocol is a message. This is a control protocol and, therefore, it does not carry payload on behalf of other layers or protocols.

This protocol uses the Signaling Protocol to transmit and receive messages.

13.2.4 Protocol initialization

13.2.4.1 Protocol initialization for the InConfiguration protocol instance

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in 15.1.1.

13.2.4.2 Protocol initialization for the InUse protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The access terminal and access network shall perform the procedures specified in 15.1.2.
- The access terminal shall set BusyCount to 0.
- The access terminal shall initialize a receive pointer for the UATI assignment validation, V(R), to 255.
- If no other Route currently exists and this is not the SessionAnchor Route, the access terminal shall perform the following:
 - 1) Set OldUATI to NULL.

- 2) Set SessionSeed to a 128-bit pseudo-random or random number.
- 3) Clear the PagingIDList
- 4) Set CurrentATI to <ATIType = '11', ATI = SessionSeed >.
- 5) Set UATI to NULL.
- 6) Set SessionSignature to 0x0000
- 7) Set RouteCounter to 0.
- The access terminal shall set LinkID to NULL.
- If this route is being opened with the access network that served as the SessionAnchor Route, the access terminal shall mark this Route as the SessionAnchor Route.
- The protocol at the access terminal and access network shall enter the WaitingToOpen State.

13.2.5 Procedures and messages for the InConfiguration instance of the protocol

13.2.5.1 Procedures

This protocol uses the Session Control Protocol to define the processing of the configuration messages.

13.2.5.2 Message formats

This protocol uses the following Session Control Protocol messages to perform attribute negotiation: ConfigurationRequest, ConfigurationResponse, FastConfigurationRequest, FastConfigurationResponse, ConfigurationCopyRequest, ConfigurationCopyAccept, and ConfigurationCopyReject.

13.2.6 Procedures and messages for the InUse instance of the protocol

13.2.6.1 Procedures

- The access terminal shall support at least six Routes (including Routes in the Radio Active Set) that are not in the InActive state.⁴⁴

13.2.6.1.1 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.3 when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

13.2.6.1.2 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in 15.1.4, in the order specified, when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

13.2.6.1.3 UATI assignment processing

Whenever the access terminal receives a UATI assignment via a UATIAssignment message, RouteOpenAccept message, or DAPAssignment message, the access terminal shall perform the following:

- Add the following entry to the PagingIDList:
<ATIType='10', PagingID> where PagingID is determined as follows:

⁴⁴Six routes are required due to the following: three Routes in Radio Active Set, one Route to perform session negotiation, one Route to the SessionAnchor access network, and one Route to add a pilot that may belong to a new access network.

- If the PagingIDInd field is set to '00', then PagingID shall be set to the value of PagingID received in the message.
- If the PagingIDInd field is set to '01', then PagingID shall be set to 32 LSB of the new UATI assigned to the access terminal, as derived below.
- If the PagingIDInd field is set to '10', then PagingID shall be set to the value of PagingID corresponding to the CurrentATI.
- The access terminal shall set CurrentATI to <ATIType = '10', ATI> where ATI is set as follows:
 - If the UATIlsbSameAsPrev field in the received message equals '1', then ATI = [ANID|64 LSB of CurrentATI] where ANID is the 64 MSB of the identifier of the access network that transmitted the message.
 - If the UATIlsbSameAsPrev field in the received message equals '0', then ATI = [ANID|UATIlsb received in the message] where ANID is the 64 MSB of the identifier of the access network that transmitted the message.
- Return a *UATIAssigned* indication to all Routes with argument containing the RouteID of this Route
- Mark this Route as the SessionAnchor Route, and shall mark all other Routes as not being the SessionAnchor Route.
- The access terminal shall send a UATIComplete message to the access network.
- The access terminal shall delete the PagingID corresponding to the previous UATI from the PagingIDList if any of the following conditions are true:
 - The UATIComplete message containing the new UATI was acknowledged
 - The access terminal has received a RouteOpenAccept on a route on which access was made with the new UATI.
 - The access terminal has performed a successful explicit registration with the new UATI.

13.2.6.1.4 LinkID procedures

- If the access terminal receives a LinkID (via the RouteOpenAccept message or the LinkIDAssignment message), the access terminal shall perform the following:
- If the current LinkID of this Route is NULL, then the Route Control Protocol shall return a *LinkInitialized* indication.
- If this LinkID is different than all of the LinkIDs of the other Routes, the access terminal shall open a new interface to the layers above the air interface.

13.2.6.1.5 DataAttachmentPoint procedures

13.2.6.1.5.1 Access terminal procedures

The access terminal shall perform the following DataAttachmentPoint Route related procedures:

- If the access terminal receives a DAPMoveRequestRequest message from the access network, the access terminal shall send a DAPMoveRequest message to the access network.
- The access terminal shall not send an unsolicited DAPMoveRequest message if ATassistedDAPMove field of the ATassistedDAPMoveMode attribute is set to 0.
- If ATassistedDAPMove field of the ATassistedDAPMoveMode attribute is set to 1,⁴⁵ the access terminal shall comply with the following requirements in determining when to send a DAPMoveRequest message to request the access network to assign this Route as the DataAttachmentPoint Route:

⁴⁵ATassistedDAPMoveMode is supported since the access terminal knows channel conditions.

- The access terminal should send a DAPMoveRequest message if all of the following conditions are satisfied, subject to other constraints of DAPMoveRequestTimer:
 - A period of L2HotoDAPMoveRequestTimer has elapsed since the Route Control Protocol has received the last ReverseControlChannelMAC.FLSSchanged indication and the FLSS public data of the Reverse Control Channel MAC Protocol correspond to this Route,
 - DFLSS and FLSS public data of the Reverse Control Channel MAC Protocol are the same, and
 - This Route is not the DataAttachmentPoint Route.
- The access terminal shall not send an unsolicited DAPMoveRequest message prior to receiving the LinkID associated with this Route.
- Upon exiting the Idle State of the AirLink Management Protocol, the access terminal shall send a DAPMoveRequest message if all of the following conditions are true:
 - The access terminal has received the LinkID associated with this Route,
 - This Route is not marked as the DataAttachmentPoint Route, and
 - The FLSS public data of the Reverse Control Channel MAC Protocol correspond to this Route.
- The access terminal shall send a DAPMoveRequest message prior to the expiration of the DataAttachmentPoint refresh timer.
- Unless specified otherwise, the access terminal shall not send an unsolicited DAPMoveRequest message if a DAPMoveRequest message was sent within the period equal to the value of the DAPMoveRequestTimer attribute.
- If the access terminal receives a DAPAssignment message from the access network with the RequestStatus field set to '1', the access terminal shall perform the following: if the Timestamp field of the message is set to a value lower than the Timestamp field of the previous DAPAssignment message (if any) processed by the access terminal, the access terminal shall discard this message; otherwise, the access terminal shall perform the following:
 - The access terminal shall mark this Route as the DataAttachmentPoint Route.
 - The Route Control Protocol shall return a *DAPAssigned(RouteID)* indication with argument set to the RouteID of this Route.
 - The access terminal shall set DAPRefreshTime local common data to the corresponding value received in the message and shall start the DataAttachmentPoint refresh timer with a value set to DAPRefreshTime.
- If this message includes a UATI assignment, the access terminal shall process the UATI assignment as specified in 13.2.6.1.3.
- If the RestartNetworkInterface field is set to '1', the access terminal shall restart the interface between the Radio Link Protocol and the higher layer. The access terminal may also restart higher layer protocols.
- If the Route Control Protocol receives a *RouteControl.DAPAssigned(RouteID)* indication from any Route, the Route Control Protocol shall perform the following: if the RouteID received with this indication is not equal to the RouteID of this Route, the Route Control Protocol shall mark this Route as not being the DataAttachmentPoint Route.

13.2.6.1.5.2 Access Network Procedures

The access network shall perform the following DataAttachmentPoint Route related procedures:

- If ATassistedDAPMove field of the ATassistedDAPMoveMode attribute is set to 1, the access network shall not send a DAPAssignment message unless in response to receiving a DAPMoveRequest message from the access terminal.

- The access network may send a *DAPMoveRequestRequest* message to request the access terminal to send a *DAPMoveRequest* message.

13.2.6.1.6 SessionAnchor procedures

13.2.6.1.6.1 Access terminal procedures

The access terminal shall perform the following SessionAnchor Route related procedures:

- If the access terminal receives a *UATIAssignment* message, the access terminal shall process the UATI assignment as specified in 13.2.6.1.3.
- If the Route Control Protocol receives a *RouteControl.UATIAssigned(RouteID)* indication, the Route Control Protocol shall perform the following:
 - If the RouteID received with this indication is not equal to the RouteID of this Route, the Route Control Protocol shall mark this Route as not being the SessionAnchor Route.

13.2.6.1.6.2 Access Network Procedures

The access network shall perform the following SessionAnchor Route related procedures:

- When the access network sends a UATI assignment via *UATIAssignment* message, *RouteOpenAccept* message, or *DAPAssignment* message, it shall perform the following:
 - Access network shall assign a Unicast access terminal Identifier (UATI) and Paging Identifier (PagingID) to the access terminal for the session.
- When the access network receives a *UATISuccess* message, the access network shall return a *UATIAssigned* indication

13.2.6.1.7 InActive State

In this state, a Route to the access network and a corresponding protocol stack does not exist. Packets cannot be exchanged over this Route. Whenever the Route Control Protocol enters the InActive State, the corresponding protocol stack shall be purged. If the SessionAnchor Route enters this state, the access terminal shall transition to the WaitingToOpen State in order to open the SessionAnchor Route.

13.2.6.1.8 WaitingToOpen State

In this state, a Route to the access network exists but is not opened. In this state, the access terminal waits for the Route to be opened. While in this state, the access terminal is allowed to send only a limited set of messages. While in this state, the access terminal may send packets on behalf of other Routes.

13.2.6.1.8.1 Access terminal requirements

While in this state, the access terminal may send a *RouteOpenRequest* message to the access network. If this is the SessionAnchor Route, then the access terminal shall send a *RouteOpenRequest* message upon entering this state.

The access terminal shall send a *RouteOpenRequest* message, if at least one of the following conditions is false:

- The access terminal entered this state for performing explicit registration
- RegistrationFailed public data of AirLink Management Protocol is set to '0'

While in this state, the access terminal shall perform the following:

- If the access terminal receives a RouteOpenRequestRequest message, the access terminal shall send a RouteOpenRequest message to this access network.
- If the access terminal sends a RouteOpenRequest message, it shall perform the following:
 - If this Route was previously open and was closed less than $T_{RCPPrevClose}$ seconds ago, the access terminal shall perform the following:
 - The Route Control Protocol shall return a RouteCreated(RouteCounter) indication, with RouteCounter set to the value transmitted in the corresponding RouteOpenRequest message.
 - The access terminal shall start the ATWaitingToOpen timer with a value of $T_{RCPATWaitOpen}$.

While in this state, the access terminal shall perform the following:

- Access terminal may transmit and receive data.
- If the access terminal receives a RouteOpenAccept message, the access terminal shall perform the following:
 - The access terminal shall add the RouteID and ANID corresponding to this Route to RouteMap.
 - The access terminal shall return a *RouteOpened* indication.
 - If RouteOpenAccept message includes a UATI assignment, the access terminal shall process the UATI assignment as specified in 13.2.6.1.3.
 - The access terminal shall transition to the Open State.
- If the access terminal receives a RouteOpenReject message, the access terminal shall perform the following:
 - If the RejectReason field is set to 0x00, the access terminal shall start a Retry Delay timer for this access network for time specified by the RetryDelay field in the message.
 - While the Retry Delay timer is running for an access network, the access terminal should not send a RouteOpenRequest message to a pilot if the access terminal determines that the pilot belongs to that access network.
 - If the access terminal receives a RouteClose message, the access terminal shall send a RouteClose message to the access network and shall transition to the InActive State.
 - If the ATWaitingToOpen timer expires, *AirLinkManagement.AirLinkInitialized* indication is received, *AirLinkManagement.IdleInitiated* indication is received, *AirLinkManagement.ConnectionClosed* indication is received, *Route.PersonalityFailure* indication is received, *Route.UATIFailed* indication is received, or *Route.RouteExists* indication is received, the access terminal shall transition to the InActive State.
- If the Route Control Protocol receives a *CreateNewStack(PilotID)* command, if the CurrentATI public data is set to RATI, the Route Control Protocol shall ignore this command; otherwise, the Route Control Protocol shall perform the following:
 - The Route Control Protocol shall select a InitialProtocolSetIdentifier or PersonalityIndex to describe the personality of the new protocol stack. If the overhead information corresponding to this pilot is available, the InitialProtocolSetIdentifier corresponding to the selected personality shall be equal to one of the InitialProtocolSetIdentifier values advertised by the access network corresponding to this pilot; otherwise, the access terminal may choose a InitialProtocolSetIdentifier or PersonalityIndex based on an implementation-defined decision.
- While in this state, the access terminal shall perform the SessionAnchor Route related procedures specified in 13.2.6.1.6.1.

13.2.6.1.8.2 Access network requirements

While in this state, the access network shall perform the following:

- Access network may transmit and receive data.
- If the access network receives a *RouteOpenRequest* message, the access network shall send a *RouteOpenAccept* message or a *RouteOpenReject* message to the access terminal.
 - If the access network sends a *RouteOpenAccept* message, the Route Control Protocol shall transition to the Open State.
 - If the access network sends a *RouteOpenReject* message, the access network shall transition to the InActiveState.
 - The Route Control Protocol shall return a *RouteCreated(RouteCounter)* indication, with *RouteCounter* set to the value received in the corresponding *RouteOpenRequest* message.
- If the access network sends a *RouteClose* message to the access terminal, the access network shall transition to the *WaitingToClose* State.
- If the access network receives a *RouteClose* message in this state, the access network shall transition to the InActive State.
- If the Route Control Protocol receives a *DeleteOldRoutes* command from a Route other than this Route, this Route Control Protocol shall transition to the InActive State.

13.2.6.1.9 Open State

In this state, a Route to the access network exists (i.e., access network has access terminal's session) and the Route is open. Packets can be exchanged over this Route.

13.2.6.1.9.1 Access terminal requirements

Upon entering this state, the access terminal shall perform the following:

- If this is not the only Route in *RouteMap*, the access terminal shall send a *RouteMap* message to the access network conveying the information in *RouteMap*.

If the Route Control Protocol receives a *CreateNewStack(PilotID)* command, if the *CurrentATI* public data is set to *RATI*, the Route Control Protocol shall ignore this command; otherwise the Route Control Protocol shall perform the following:

- The Route Control Protocol shall select a *InitialProtocolSetIdentifier* or *PersonalityIndex* to describe the personality of the new protocol stack. The *InitialProtocolSetIdentifier* corresponding to the selected personality shall be equal to one of the *InitialProtocolSetIdentifier* values advertised by the access network corresponding to this pilot.
- The Route Control Protocol shall create a new Route (protocol stack) with the selected *InitialProtocolSetIdentifier* or *PersonalityIndex*.

If the access terminal receives a *RouteCreate* message, the access terminal shall perform the following:

- The Route Control Protocol shall select a *ProtocolSetIdentifier* or *PersonalityIndex* from the list included in the message to describe the Personality of the Route.
- The Route Control Protocol shall create a new Route (protocol stack) with the selected Personality.

If the Route Control Protocol receives a *RouteControl.RouteOpened* indication or a *RouteControl.RouteClosed* indication from another route, the Route Control Protocol shall send *RouteMap* message to the access network conveying the information in *RouteMap*, unless the State public data of the Airlink Management Protocol is set to *Idle*.

If the Route Control Protocol receives a *Signaling.ReliableSignalingStreamReset* indication and this is the SessionAnchor Route, the access terminal shall transition to the InActive State.

If the access terminal receives a RouteMapRequest message, the access terminal shall send a RouteMap message to the access network conveying the information in RouteMap.

If an *AirLinkManagement.AirLinkInitialized* indication is received or an *AirLinkManagement.IdleInitiated* indication is received, and this is not the SessionAnchor Route, the access terminal shall transition to the InActive State, and the access terminal shall remove the RouteID and ANID corresponding to this Route from RouteMap.

Access terminal may transmit and receive data.

The access terminal shall perform the DataAttachmentPoint Route related procedures specified in 13.2.6.1.5.

The access terminal shall perform the LinkID related procedures specified in 13.2.6.1.4.

The access terminal shall perform the SessionAnchor Route related procedures specified in 13.2.6.1.6.

If the access terminal receives a RouteClose message in this state, the access terminal shall perform the following:

- The access terminal shall send a RouteClose message to the access network.
- The access terminal shall remove the RouteID and ANID corresponding to this Route from the RouteMap.
- The access terminal shall return a *RouteClosed* indication and shall transition to the InActive State.

If the access terminal receives a *CloseRoute* command in this state, the access terminal shall perform the following:

- The access terminal shall send a RouteClose message to the access network.
- The access terminal shall remove the RouteID and ANID corresponding to this Route from the RouteMap.
- The access terminal shall return a *RouteClosed* indication and shall transition to the InActive State.

If all of the following conditions are true, the access terminal shall start the ATRouteClose timer with a value of ATRouteCloseTimer:

- State public data of the AirLink Management Protocol is not set to “Connected.”
- This is not the SessionAnchor Route.

The ATRouteClose timer shall be disabled if any of the above conditions is no longer true.

- Unless the access terminal receives a RouteClose message or a *CloseRoute* command, the access terminal shall comply with the following requirements in determining when to send a RouteClose message: The access terminal shall not send a RouteClose message if this is the SessionAnchor Route.
- This is the DataAttachmentPoint Route and the value of the ATAssistedDAPMove field of the ATAssistedDAPMoveMode attribute is equal to 1.
 - State public data of the AirLink Management Protocol is set to “Connected.”
- The ATRouteClose timer is active.
- The access terminal shall not send a RouteClose message if the ATRouteClose timer is active.

- The access terminal should not send a RouteClose message if the Busy public data of any protocol of this Route is not zero.

If the access terminal sends a RouteClose message, the access terminal shall perform the following:

- The access terminal shall remove the RouteID and ANID corresponding to this Route from the RouteMap.
- The access terminal shall return a *RouteClosed* indication and shall transition to the InActive State.

13.2.6.1.9.2 Access network requirements

If the access network receives a *CloseRoute* command in this state, the access network shall perform the following:

- The access network shall send a RouteClose message to the access terminal.
- The access network shall transition to the WaitingToClose State.

The access network may query the current route list by sending a RouteMapRequest message.

If the access network receives a RouteMap message in this state, the access network shall send a RouteMapAck message to the access terminal.

If the access network receives a RouteClose message in this state, the access network shall transition to the InActive State.

If the access network receives a RouteOpenRequest message, the access network shall send a RouteOpenAccept message.

The access network shall perform the DataAttachmentPoint Route related procedures specified in 13.2.6.1.5.2.

The access network shall perform the SessionAnchor Route related procedures specified in 13.2.6.1.6.22.

The access network should send a RouteClose message if all of the following conditions are true:

- The State public data of the AirLink Management Protocol is not set to “Connected.”
- If the Busy public data of all protocols of this Route is zero.
- This is not the SessionAnchor Route.

If the Route Control Protocol receives a DeleteOldRoutes command from a Route other than this Route, this Route Control Protocol shall transition to the InActive State.

If the Route Control Protocol receives *QueryRouteMap* command, the access network shall send a RouteMapRequest message to the access terminal.

13.2.6.1.10 WaitingToClose State

This state applies only to the access network. In this state the access network waits for the Route close procedures to complete.

Upon entering this state, the access network shall start the ANRouteClose timer with a value of $T_{RCPANRouteClose}$.

While in this state, if the access network receives a RouteClose message from the access terminal or the ANRouteClose timer expires, the access network shall return a *RouteClosed* indication and shall transition to the InActive State.

If the access network receives a *Route.RouteReopen* indication, the Route Control Protocol shall terminate this Route and open a new one.

If the Route Control Protocol receives a DeleteOldRoutes command from a Route other than this Route, this Route Control Protocol shall transition to the InActive State.

13.2.6.2 Message formats

13.2.6.2.1 RouteOpenRequest

The access terminal sends the RouteOpenRequest message to request opening a Route (see Table 357).

Table 357—RouteOpenRequest message

Field	Length (bits)
MessageID	8
RouteCounter	15
Reserved1	1
SessionSignature	16
RouteOpenRequestReason	4
EmergencyIndication	1
AuthenticationTagIncluded	1
Reserved2	2 ^a
AuthenticationTagBlobLength	8 ^a
AuthenticationTagBlob	8 × AuthenticationTagBlobLength
SessionAnchorRouteID	7 ^a
Padding	0–2 (as needed)

^aThis field may be absent (see field description for conditions).

MessageID	The access terminal shall set this field to 0x00.
RouteCounter	The access terminal shall increment this field to the next available value each time a RouteOpenRequest message is sent, where the next available value is the value whose seven least significant bits are not equal to the RouteID of an existing Route. The seven least significant bits of this field shall be used as the RouteID for this Route.
Reserved1	The access network shall ignore this field.
SessionSignature	The access terminal shall set this field to the signature of the current session.

RouteOpenRequestReason

The access terminal shall set this field to the reason for Route open request as specified in Table 358.

Table 358—RouteOpenRequestReason encoding

RejectReason	Meaning
0x0	AT-initiated
0x1	Page Response
0x2	Registration Failed
0x3	Response to RouteCreate message
0x4	Response to RouteOpenRequestRequest message
All other values	Reserved

EmergencyIndication The access terminal shall set this field to '1' if this is an emergency RouteOpenRequest; Otherwise, the access terminal shall set this field to '0'.

AuthenticationTagIncl The access terminal shall set this field to '0' if the CurrentATI public data is set to RATI or this is the SessionAnchor Route; otherwise, the access terminal shall set this field to '1'.

Reserved2 The access terminal shall omit this field if the AuthenticationTagIncl field is set to '0'. The access network shall ignore this field.

AuthenticationTagBlobLength

The access terminal shall omit this field if the AuthenticationTagIncluded field is set to '0'; otherwise, the access terminal shall include this field and set it as follows:

The access terminal shall set this field to the length of the AuthenticationTagBlob field in units of octets that is equal to value of IDTagSize returned by the CREATE_ID_TAG(SequenceNumber) procedure call to the Message Integrity Protocol of the SessionAnchor Route, where SequenceNumber is set to the 32-bit value (17-bit of '0's | RouteCounter).

AuthenticationTagBlob The access terminal shall omit this field if the AuthenticationTagIncluded field is set to '0'; otherwise, the access terminal shall include this field and set it as follows:

The access terminal shall set this field to the value of IDTag returned by the CREATE_ID_TAG(SequenceNumber) procedure call to the Message Integrity Protocol of the SessionAnchor Route, where SequenceNumber is set to the 32-bit value (17-bit of '0's | RouteCounter).

SessionAnchorRouteID The access terminal shall omit this field if the AuthenticationTagIncluded field is set to '0'; otherwise, the access terminal shall include this field and set it to the Route Identifier of the SessionAnchor Route.

Padding The access terminal shall add padding bits to make the length of the entire message equal to an integer number of octets. The access terminal shall set these bits to '0'. The access network shall ignore this field.

Channels	RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.2 RouteOpenAccept

The access network sends the RouteOpenAccept message to accept the access terminal request for opening a new Route (see Table 359).

Table 359—RouteOpenAccept message

Field	Length (bits)
MessageID	8
ANIDmsb	64
LinkIDIncluded	1
LinkID	64 ^a
MTU	8 ^a
UATIAssignIncl	1
UATIlsbSameAsPrev	1 ^a
UATIlsb	64 ^a
PagingIDInd	2 ^a
PagingID	32 ^a
Padding	0–7 (as needed)

^aThis field may be absent (see field description for conditions).

MessageID	The access network shall set this field to 0x01.
ANIDmsb	The access network shall set this field to the 64 MSB of its access network identifier.
LinkIDIncluded	The access network shall set this field to '1' if the LinkID field is included in this message.
LinkID	The access network shall omit this field if the LinkIDIncluded field is set to '0'; otherwise, the access network shall include this field and set it to its link identifier.
MTU	The access network shall omit this field if the LinkIDIncluded field is set to '0'; otherwise, the access network shall include this field and set it such that the maximum transmission unit of IP datagrams over this Route (in octets) equals 1280 + MTU.

UATIAssignIncl	The access network shall set this field to '1' if UATI and PagingID assignment is included in this message; otherwise, the access network shall set this field to '0'.
UATIlsbSameAsPrev	The access network shall omit this field if UATIAssignIncl field is set to '0'; otherwise, the access network shall include this field and set it as follows: The access network shall set this field to '1' if the 64 LSB of the UATI being assigned is the same as the 64 LSB of the UATI currently assigned to the access terminal; otherwise, the access network shall set this field to '0'.
UATIlsb	If the UATIlsbSameAsPrev field is not included or is set to '1', the access network shall omit this field; otherwise, the access network shall include this field and set it to the 64 LSB of the UATI being assigned to the access terminal.
PagingIDInd	The access network shall omit this field if UATIAssignIncl field is set to '0'; otherwise, the access network shall include this field and set it as follows: The access network shall set this field to '00' if the PagingID field is included in this message. The access network shall set this field to '01' if PagingID corresponds to the 32 LSB of the UATI assigned to the access terminal. The access network shall set this field to '10' if the PagingID is the same as the current value of PagingID.
PagingID	The access network shall omit this field if the PagingIDInd field is not set to '00'; otherwise, the access network shall include this field and set it to the unicast paging identifier being assigned to the access terminal.
Padding	The access network shall add padding bits to make the length of the entire message equal to an integer number of octets. The access network shall set these bits to '0'. The access terminal shall ignore this field.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.3 RouteClose

The sender sends the RouteClose message to close the current Route (see Table 360).

Table 360—RouteClose message

Field	Length (bits)
MessageID	8

MessageID The sender shall set this field to 0x02.

Channels	FTC RTC	RLP	Reliable Best effort (FL only)
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.4 DAPMoveRequest

The access terminal sends the DAPMoveRequest message to request changing the data anchor access network to the access network corresponding to the Route that generated this message (see Table 361).

Table 361—DAPMoveRequest message

Field	Length (bits)
MessageID	8

MessageID

The access terminal shall set this field to 0x03.

Channels	RTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key is available

13.2.6.2.5 DAPAssignment

The access network sends the DAPAssignment message to assign a new data anchor access network to the access terminal (see Table 362).

Table 362—DAPAssignment message

Field	Length (bits)
MessageID	8
RestartNetworkInterface	1
RequestStatus	1
Reserved1	6
TimeStamp	16 ^a
DAPRefreshTime	16 ^a
UATIAssignIncl	1 ^a
UATIIsbSameAsPrev	1 ^a
UATIIsb	64 ^a
PagingIDInd	2 ^a
PagingID	32 ^a
Padding	0–7 (as needed)

^aThis field may be absent (see field description for conditions).

MessageID

The access network shall set this field to 0x04.

RestartNetworkInterface

If the access network wants to request the access terminal to restart the network interface, the access network shall set this field to '1'; otherwise, the access network shall set this field to '0'.

RequestStatus	If this message is being sent in response to a DAPMoveRequest message from the access terminal and the access network does not accept the DataAttachmentPoint move request, the access network shall set this field to '0'; Otherwise, the access network shall set this field to '1'.
Reserved1	The access terminal shall ignore this field.
Timestamp	The access network shall omit this field if the RequestStatus field is set to '0'; otherwise, the access network shall include this field and set it as follows: The access network shall set this field to the system time at which this message was generated, specified in the NTP timestamp format [RFC1305].
DAPRefreshTime	The access network shall omit this field if the RequestStatus field is set to '0'; otherwise, the access network shall include this field and set it as follows: The access network shall set this field to the refresh time, in seconds, associated with this DataAttachmentPoint assignment. The access network shall set this field to 0xffff to indicate a timer value of infinity.
UATIAssignIncl	The access network shall omit this field if the AcceptInd field is set to '0'; otherwise, the access network shall include this field and set it to '1' if UATI and PagingID assignment is included in this message; otherwise, the access network shall set this field to '0'.
UATIlsbSameAsPrev	The access network shall omit this field if UATIAssignIncl field is set to '0'; otherwise, the access network shall include this field and set it as follows: The access network shall set this field to '1' if the 64 LSB of the UATI being assigned is the same as the 64 LSB of the UATI currently assigned to the access terminal; otherwise, the access network shall set this field to '0'.
UATIlsb	If the UATIlsbSameAsPrev field is not included or is set to '1', the access network shall omit this field; otherwise, the access network shall include this field and set it to the 64 LSB of the UATI being assigned to the access terminal.
PagingIDInd	The access network shall omit this field if UATIAssignIncl field is set to '0'; otherwise, the access network shall include this field and set it as follows: The access network shall set this field to '00' if the PagingID field is included in this message. The access network shall set this field to '01' if PagingID corresponds to the 32 LSB of the UATI assigned to the access terminal. The access network shall set this field to '10' if the PagingID is the same as the current value of PagingID.
PagingID	The access network shall omit this field if the PagingIDInd field is not set to '00'; otherwise, the access network shall include this field and set it to the unicast paging identifier being assigned to the access terminal.

Padding The access network shall add padding bits to make the length of the entire message equal to an integer number of octets. The access network shall set these bits to '0'. The access terminal shall ignore this field.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.6 RouteMap

The access terminal sends the RouteMap message to provide the mapping of short Route identifier (RouteID) to long Route identifier (ANID) of all existing Routes in RouteMap (see Table 363).

Table 363—RouteMap message

Field	Length (bits)
MessageID	8
TransactionID	8
Reserved1	4
NumRoutes	4
NumRoutes occurrences of the following record:	
Reserved2	1
RouteID	7
ANIDmsb	64

MessageID The access terminal shall set this field to 0x05.

TransactionID The sender shall set this field to zero if this is the first instance of this message sent by the InUse instance, and shall increment the field (modulo 256) for each successive message. In this protocol, all messages share a common TransactionID counter, so that the ConfigurationReject message can uniquely identify one of four types of rejected messages.

Reserved1 The access network shall ignore this field.

NumRoutes The access terminal shall set this field to the number of Routes in RouteMap.

Reserved2 The access network shall ignore this field.

RouteID The access terminal shall set this field to the RouteID assigned to the Route. The access terminal shall not change the RouteID assigned to this Route.

ANIDmsb The access terminal shall set this field to the 64 MSB of the access network identifier corresponding to this Route.

Channels	RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.7 RouteMapAck

The access network sends the RouteMapAck message to acknowledge the reception of a RouteMap message from the access terminal (see Table 364).

Table 364—RouteMapAck message

Field	Length (bits)
MessageID	8
TransactionID	8

- MessageID

The access network shall set this field to 0x06.
- TransactionID

The access network shall set this field to the value of the TransactionID field in the RouteMap message to which this message is the acknowledgement.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.8 RouteMapRequest

The access network sends the RouteMapRequest message to query the access terminal for the current RouteMap (see Table 365).

Table 365—RouteMapRequest message

Field	Length (bits)
MessageID	8

- MessageID

The access network shall set this field to 0x07.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.9 UATIAssignment

The access network sends the UATIAssignment message to assign a UATI to the access terminal.

- MessageID

The access network shall set this field to 0x08.
- UATILsbSameAsPrev

The access network shall set this field to '1' if the 64 LSB of the UATI being assigned is the same as the 64 LSB of the UATI currently assigned to the access terminal; otherwise, the access network shall set this field to '0'.
- UATILsb

If the UATILsbSameAsPrev field is set to '1', the access network shall omit this field; otherwise, the access network shall include this field and set it to the 64 LSB of the UATI being assigned to the access terminal (see Table 366).

Table 366—UATIAssignment message

Field	Length (bits)
MessageID	8
UATIlsbSameAsPrev	1
UATIlsb	64 ^a
PagingIDInd	2
PagingID	32 ^a
Reserved	5

^aThis field may be absent (see field description for conditions).

PagingIDInd The access network shall set this field to '00' if the PagingID field is included in this message. The access network shall set this field to '01' if PagingID corresponds to the 32 LSB of the UATI assigned to the access terminal. The access network shall set this field to '10' if the PagingID is the same as the current value of PagingID.

PagingID The access network shall omit this field if the PagingIDInd field is not set to '00'; otherwise, the access network shall include this field and set it to the unicast paging identifier being assigned to the access terminal.

Reserved This field shall be ignored by the receiver.

Channels	FTC
Addressing	Unicast

RLP	Best effort
AuthTag	Required if key is available

13.2.6.2.10 UATIComplete

The access terminal sends this message to notify the access network that it has received the UATIAssignment message (see Table 367).

Table 367—UATIComplete message

Field	Length (bits)
MessageID	8
RouteID	7
UATIlsbSameAsPrev	1
UATIlsb	64 ^a
PagingIDInd	2
PagingID	32 ^a
Reserved	6

^aThis field may be absent (see field description for conditions).

MessageID	The access terminal shall set this field to 0x09.
RouteID	The access terminal shall set this field to the RouteID of the route that performed the UATI assignment.
UATIlsbSameAsPrev	The access terminal shall set this field to '1' if the 64 LSB of the UATI that was assigned is the same as the 64 LSB of the UATI previously assigned to the access terminal; otherwise, the access terminal shall set this field to '0'.
UATIlsb	If the UATIlsbSameAsPrev field is set to '1', the access terminal shall omit this field; otherwise, the access terminal shall include this field and set it to the 64 LSB of the UATI that was assigned to the access terminal.
PagingIDInd	The access terminal shall set this field to '00' if the PagingID field is included in this message. The access terminal shall set this field to '01' if PagingID corresponds to the 32 LSB of the UATI conveyed in this message. The access terminal shall set this field to '10' if the PagingID is the same as the previous value of PagingID.
PagingID	The access terminal shall omit this field if the PagingIDInd field is not set to '00'; otherwise, the access terminal shall include this field and set it to the unicast paging identifier assigned to the access terminal.
Reserved	This field shall be ignored by the receiver.

Channels	RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.11 RouteCreate

The access network sends the RouteCreate message to request the access terminal to open a Route to the access network.⁴⁶ (See Table 368.)

Table 368—RouteCreate message

Field	Length (bits)
MessageID	8
ANIDmsb	64
Reserved1	5
NumProtocolSetIdentifier	3
NumProtocolSetIdentifier occurrences of the following field:	
ProtocolSetIdentifier	16
NumPersonalityIndex	3
Reserved2	5

⁴⁶This can be used by an access network that intends to become the SessionAnchor.

Table 368—RouteCreate message (continued)

Field	Length (bits)
NumPersonalityIndex occurrences of the following field:	
PersonalityIndex	4
Padding	0–7 (as needed)

MessageID	The access network shall set this field to 0x0a.
ANIDmsb	The access network shall set this field to the access network identifier corresponding to the 64 MSB of the access network to which a Route is to be created.
Reserved1	The access terminal shall ignore this field.
NumProtocolSetIdentifier	The access network shall set this field to the number of ProtocolSetIdentifier fields included in this message.
ProtocolSetIdentifier	The access network shall set this field to the ProtocolSetIdentifier corresponding to the personality of the this Route.
NumPersonalityIndex	The access network shall set this field to the number of PersonalityIndex fields included in this message.
Reserved2	The access terminal shall ignore this field.
PersonalityIndex	The access network shall set this field to the PersonalityIndex corresponding to a personality allowed for this Route. The access network shall list these fields in decreasing order of preference.
Padding	The access network shall add padding bits to make the length of the entire message equal to an integer number of octets. The access terminal shall ignore this field.

Channels	FTC
Addressing	Unicast

RLP	Reliable
AuthTag	Required if key is available

13.2.6.2.12 RouteOpenReject

The access network sends the RouteOpenReject message to reject a RouteOpenRequest message from the access terminal (see Table 369).

Table 369—RouteOpenReject message

Field	Length (bits)
MessageID	8
RejectReason	8

Table 369—RouteOpenReject message (continued)

Field	Length (bits)
RetryDelay	8 ^a

^aThis field may be absent (see field description for conditions).

MessageID	The access network shall set this field to 0x0b.
RejectReason	The access network shall set this field to the reason for rejection as specified in Table 370.

Table 370—RejectReason encoding

RejectReason	Meaning
0x00	Route Open Request queued
0x01	Session fetch failed due to session not found
0x02	Session fetch failed due to authentication failure
0x03	General authentication/billing failure/ Other
0x04	No network connectivity
0x05	System overloaded
0x06	Pilot belongs to existing Route
All other values	Reserved

RetryDelay	The access network shall omit this field if the RejectReason field is not set to 0x00; otherwise, the access network shall include this field and set it to the retry delay in units of seconds as follows:

Let $a = \text{RetryDelay}[4:0]$ and $b = \text{RetryDelay}[7:5]$, the effective retry delay is $2^b \times a$ in units of seconds.

Channels	FTC	RLP	Reliable
Addressing	Unicast	AuthTag	Required if key is available

13.2.6.2.13 LinkIDAssignment

The access network sends the LinkIDAssignment message to convey its link identifier to the access terminal (see Table 371).

Table 371—LinkIDAssignment message

Field	Length (bits)
MessageID	8

Table 371—LinkIDAssignment message (continued)

Field	Length (bits)
LinkID	64
MTU	8

MessageID The access network shall set this field to 0x0c.

LinkID The access network shall set this field to its link identifier.

MTU The access network shall include this field and set it such that the maximum transmission unit of IP datagrams over this route (in octets) equals 1280+MTU.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key is available

13.2.6.2.14 DAPMoveRequestRequest

The access network sends the DAPMoveRequestRequest message to request the access terminal to send a DAPMoveRequest message (see Table 372).

Table 372—DAPMoveRequestRequest message

Field	Length (bits)
MessageID	8

MessageID The access network shall set this field to 0x0d.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key is available

13.2.6.2.15 RouteOpenRequestRequest

The access network sends the RouteOpenRequestRequest message to request the access terminal to send a RouteOpenRequest message for the same Route (see Table 373).

Table 373—RouteOpenRequestRequest message

Field	Length (bits)
MessageID	8

MessageID The access network shall set this field to 0x0e.

Channels	FTC

RLP	Reliable

Addressing	Unicast

AuthTag	Required if key is available

13.2.6.3 Interface to other protocols

13.2.6.3.1 Commands Sent

This protocol does not issue any commands.

13.2.6.3.2 Indications

This protocol registers to receive the following indications:

- *AirLinkManagement.AirLinkInitialized*
- *AirLinkManagement.IdleInitiated*
- *AirLinkManagement.ConnectionClosed*
- *RouteControl.DataAttachmentPointAssigned* (from other routes)
- *RouteControl.UATIAssigned* (from other routes)
- *RouteControl.RouteOpened* (from other routes)
- *RouteControl.RouteClosed* (from other routes)
- *Route.PersonalityFailure*
- *Route.RouteExists*
- *Route.RouteReopen*
- *Route.UATIFailed*
- *Signaling.ReliableSignalingStreamReset*

13.2.7 Configuration attributes

13.2.7.1 Simple attributes

The negotiable simple attributes for this protocol are listed in Table 374. The access terminal shall use as defaults the values in Table 374 typed in ***bold italics***.

Table 374—Configurable attributes

Attribute ID	Attribute	Commit/Scope	Values	Meaning
0x0000	ATRouteCloseTimer	Soft/Static	<i>0x0005</i>	Default route close timer
			0x0001–0x0004	Route close timer period in seconds
			All other values	Reserved

13.2.7.2 Complex attributes

13.2.7.2.1 ATassistedDAPMoveMode attribute

The sender shall set AttributeID field to 0x8000.

Table 375—ATassistedDAPMoveMode attribute

Field	Length (bits)	Default
ATassistedDAPMove	1	1

Table 375—ATassistedDAPMoveMode attribute (continued)

Field	Length (bits)	Default
DAPMoveRequestTimer	15	0x0000
L2HotoDAPMoveRequestTimer	15	0x0000
Padding	1	N/A

ATassistedDAPMove The sender shall set this field to 1 if the access terminal is allowed to send a DAPMoveRequest message.

DAPMoveRequestTimer The sender shall set this field to Recommended minimum period, in units of 10ms, between unsolicited DAPMoveRequest messages. The sender shall not set this field to a value greater than 0x03e8.

L2HotoDAPMoveRequestTimer
The sender shall set this field to the interval, in units of 10ms, for which the access terminal must wait after detecting a L2 handoff before sending a DAPMoveRequest message. The sender shall not set this field to a value greater than 0x03e8.

Padding The sender shall set this bit to '0' to make the length of the entire message equal to an integer number of octets. The access terminal shall ignore this field.

Commit	Soft

Scope	Static

13.2.8 Non-attribute data

13.2.8.1 PagingID

The sender shall set the DataID to 0x0000 (see Table 376).

Table 376—PagingID

Field	Length (bits)
NumPagingID	3
NumPagingID occurrences of the following record	
PagingIDType	2
PagingID	32
Padding	0–7 (as needed)

NumPagingID This field shall be set to the number of PagingID records included.

PagingIDType This field shall be set to the type of the included PagingID as follows: This field shall be set to '01' for Multicast PagingID and '10' for Unicast PagingID. No more than one occurrence of PagingID of each PagingIDType shall be included.

PagingID This field shall be set to the current paging identifier assigned to the access terminal.

Padding This field shall be set to reserved bits to make the length of the entire record equal to an integer number of octets. These bits shall be set to '0'.

13.2.8.2 SessionSeed

The sender shall set the DataID to 0x0001 (see Table 377).

Table 377—SessionSeed

Field	Length (bits)
SessionSeed	128

SessionSeed This field shall be set to the value of the SessionSeed associated with the access terminal's session.

13.2.9 Protocol numeric constants

Protocol numeric constants are listed in Table 378.

Table 378—Protocol numeric constants

Constant	Meaning	Value
$N_{RCPType}$	Type field for this protocol	See 16.1
$N_{RCPBasic}$	Subtype field for this protocol	0x0000
$T_{RCPATWaitOpen}$	Duration is seconds for access terminal to wait in the WaitingToOpen state for a RouteOpenAccept message from the access network	1
$T_{RCPANRouteClose}$	Duration is seconds for access network to wait, after sending a RouteClose message, for a RouteClose message from the access terminal	1

13.2.10 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records.

All configuration attributes are Session State Information records. This protocol does not define additional parameter records.

14. Broadcast support

14.1 Introduction

- Broadcast-Multicast Upper Layer consists of following protocols:
- Broadcast Control Protocol
- Broadcast Packet Consolidation Protocol
- Broadcast Security Protocol
- Broadcast Inter-Route Tunneling Protocol
- Broadcast MAC Protocol

14.2 Basic Broadcast Protocol Suite

14.2.1 Overview

The broadcast packet data system provides a packet stream that can be used to carry higher-layer packets from the access network to multiple access terminals. The Forward Broadcast Multicast Services Channel carries packets containing content generated by a content server. The Forward Broadcast Multicast Services Channel can carry Forward Link signaling messages generated by the Broadcast Protocol Suite. The Broadcast Channel can also carry payload from other Routes. The Forward Broadcast Multicast Services Channel has a Forward Link, but does not have a Reverse Link. Forward Link messages are sent for transmission directly on the Forward Broadcast Multicast Services Channel or are tunneled through the Inter-Route Tunneling Protocol of a unicast Route. Reverse Link messages are tunneled through the Inter-Route Tunneling Protocol of a unicast Route. The Forward Broadcast Multicast Services Channel consists of Broadcast Physical Channels and Broadcast Logical Channels. Broadcast-Multicast Flows (also called BCMCS Flows) as well as the signaling messages destined to the Forward Broadcast Multicast Services Channel are associated with Broadcast Logical Channels and are transmitted over Broadcast Physical Channels.

The Broadcast Physical Channels consist of several sub-channels called interlace-multiplex pairs. The structure of the interlace-multiplex pairs may be different across sectors. The Basic Broadcast Protocol Suite specifies a Broadcast MAC Protocol and a Broadcast Physical Layer Protocol, which describe the structure of Broadcast Physical Channels.

A Broadcast Logical Channel (also called a logical channel) refers to a set of one or more interlace-multiplex pairs of the Broadcast Physical Channel associated with a sector over which broadcast content is transmitted. Each logical channel carries one or more BCMCS Flows. An interlace-multiplex pair associated with a sector can be assigned to at most one logical channel.

A logical channel is identified by a pair of the form (sector, BCIndex) where a sector is identified by the pair (SectorId, BCMCS Channel). BCMCS Channel refers to the frequency assignment associated with a single Channel. BCIndex refers to the value corresponding to the first PHY frame, among the set of all PHY frames of the set of interlace-multiplex pairs associated with the logical channel, which occurs on or after frame index zero.

A Broadcast-Multicast Service Flow identifier (BCMCSFlowID) identifies a Broadcast-Multicast flow (also called a BCMCS Flow). The content of a given BCMCS Flow may change with time. A BCMCS Flow is analogous to a single multimedia flow. The contents of a BCMCS Flow are not divided across multiple logical channels.

The Basic Broadcast Protocol Suite contains the following protocols:

- *Broadcast Control Protocol*: The Broadcast Control Protocol defines procedures used to control various aspects of the operation of the broadcast packet data system, such as BCMCS Flow registration requirements. The Broadcast Control Protocol also defines the BroadcastParameters message. The Broadcast Control Protocol is defined in 14.2.2.
- *Broadcast Inter-Route Tunneling Protocol*: The Broadcast Inter-Route Tunneling Protocol performs tunneling of packets generated by the unicast Routes on the Broadcast Physical Channel. The Broadcast Inter-Route Tunneling Protocol is defined in 14.2.5.
- *Broadcast Packet Consolidation Protocol*: The Broadcast Packet Consolidation Protocol performs framing of higher-layer packets and multiplexes higher-layer packets and signaling messages. The Broadcast Packet Consolidation Protocol is defined in 14.2.3.
- *Broadcast Security Protocol*: The Broadcast Security Protocol provides encryption of Broadcast Packet Consolidation Protocol payload. The Broadcast Security Protocol is defined in 14.2.4.
- *Broadcast MAC Protocol*: The Broadcast MAC Protocol defines procedures used to transmit via the Forward Broadcast Multicast Services Channel. The Broadcast MAC Protocol also provides Forward Error Correction (FEC) and multiplexing to reduce the radio link error rate as seen by the higher layers. The Broadcast MAC Protocol is defined in 14.2.6.
- *Broadcast Physical Layer Protocol*: The Broadcast Physical Layer Protocol provides the channel structure for the Forward Broadcast Multicast Services Channel. The Broadcast Physical Layer Protocol is defined in 9.5.

The relationship between the protocols in the Basic Broadcast Protocol Suite is shown in Figure 162.

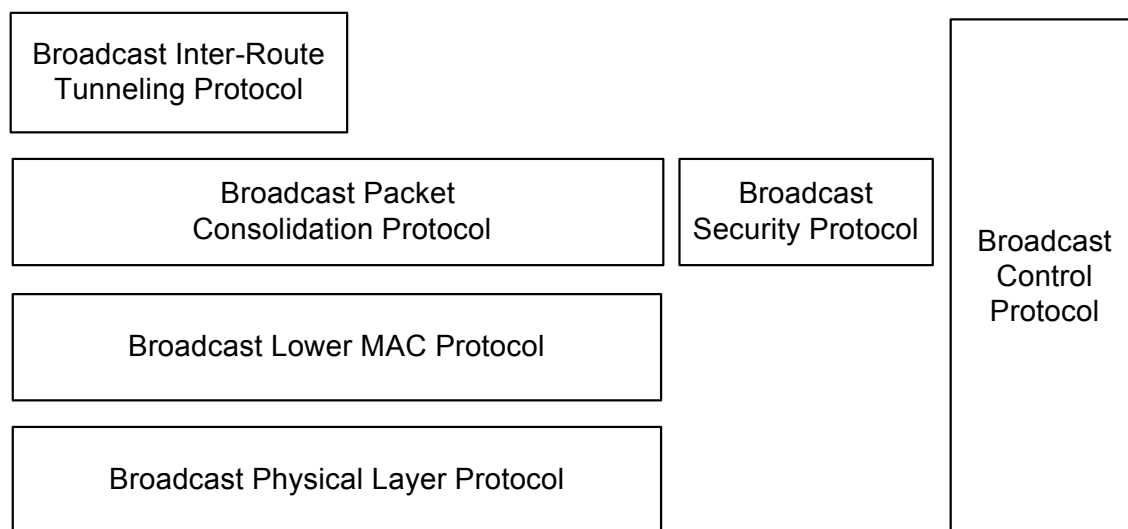


Figure 162—Basic Broadcast Protocol suite

The Broadcast Lower MAC Protocol described in Clause 4 and the Broadcast Physical Layer Protocol described in Clause 5 specifies the rules governing the transmission of logical channels on the Forward Broadcast Multicast Services Channel.

The data path for each type of packets handled by the Basic Broadcast Protocol Suite is shown in Figure 163.

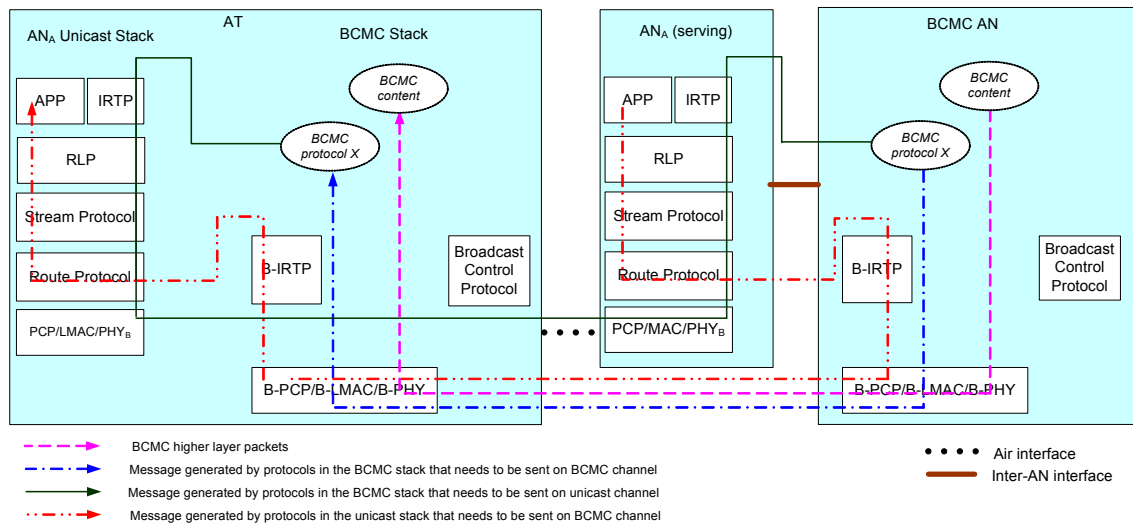


Figure 163—Data Path in Basic Broadcast Protocol suite

14.2.1.1 Basic protocol numbers

The Subtype field for the Basic Broadcast Protocol Suite is two octets, set to $N_{\text{BroadcastBasicSubtype}}$.

14.2.1.2 Primitives, local common data, and public data

14.2.1.2.1 Commands

This protocol does not define any commands.

14.2.1.2.2 Return indications

This protocol does not return any indications.

14.2.1.2.3 Local common data

This protocol does not define any local common data.

14.2.1.2.4 Public data

This protocol shall make the following data public:

- Subtype for the Basic Broadcast Protocol Suite
- Contents of the BroadcastParameters message.

14.2.1.3 Messages

Messages generated by all the protocols in the Basic Broadcast Protocol Suite are always an integer number of octets in length. The sender shall include reserved bits at the end of the message, if necessary, to make the message an integer number of octets in length. The sender shall set the reserved bits to '0'. The receiver shall ignore the value of the reserved bits.

14.2.1.4 Basic Broadcast Protocol suite initialization

14.2.1.4.1 Protocol initialization for the InConfiguration protocol instance

This protocol does not define an InConfiguration instance.

14.2.1.4.2 Protocol initialization for the InUse Protocol instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the following:

- The Broadcast Protocol Stack shall be created with a personality corresponding to the value of the InitialProtocolSetIdentifier parameter.

14.2.1.5 Procedures and messages for the InConfiguration instance of the Basic Broadcast Protocol suite

This protocol does not define an InConfiguration instance.

14.2.1.6 Procedures and messages for the InUse instance of the protocol

14.2.1.6.1 Procedures

The procedures associated with the InUse instance of this protocol is specified within each of the sub-protocols of the Basic Broadcast Protocol Suite.

14.2.1.6.1.1 Command processing

The list of events that causes an *Activate* or *Deactivate* command to be sent to this protocol is outside the scope of this standard.

14.2.1.6.2 Message formats

The messages associated with the InUse instance of this protocol is specified within each of the sub-protocols of the Basic Broadcast Protocol Suite.

14.2.2 Broadcast Control Protocol

14.2.2.1 Overview

The access terminal is required to notify the access network of the BCMCS Flows that the access terminal is interested in monitoring. This is accomplished using BCMCS Flow registration. The Broadcast Control Protocol provides procedures related to BCMCS Flow registration.

14.2.2.2 Protocol data unit

The transmission unit of this protocol is a message. This is a control protocol; and, therefore, it does not carry payload on behalf of other layers or protocols.

14.2.2.3 Procedures

14.2.2.3.1 Structure of the BCMCS Flow Identifier

A BCMCSFlowID identifies each Broadcast-Multicast flow. The BCMCSFlowID may be 16, 24, or 32 bits long.

14.2.2.3.2 Access network requirements

In order for the access terminal to discover and monitor broadcast content successfully, various broadcast-related parameters need to be signaled over the air interface. The access network broadcasts these parameters over the Control Channel in the form of BroadcastParameters message.

The BroadcastParameters message transmitted by a sector contains the logical to physical channel mapping information for that sector. A BroadcastParameters message transmitted with the CompleteListInd field set to '1' includes all available BCMC flows; a BroadcastParameters message transmitted with the CompleteListInd field set to '0' includes only those BCMC flows added since the last BroadcastParameters was transmitted.

The access network shall comply with the following requirements when transmitting BroadcastParameters message CompleteListInd field set to '1':

- The access network shall transmit the BroadcastParameters message with the broadcast MACID as a unicast message over every superframe j such that $j \bmod N_{\text{BCIPeriod}} = N_{\text{BCIPeriod}} - 1$.
- The location of this message shall be between the PHY frames 2 and 7 inclusive of the corresponding superframe.

The access network shall comply with the following requirements when transmitting BroadcastParameters message CompleteListInd field set to '0':

The access network shall transmit the BroadcastParameters message only in a superframe with an odd superframe index.

The access network may send QPCH block to notify the targeted access terminals before sending a BroadcastParameters message CompleteListInd field set to '0'.

The access network may use timer-based BCMCS Flow registration to keep track of which BCMCS Flows the access terminal may be monitoring, so that the access network may send messages to the access terminal using the appropriate channel(s). The access network may use the BCMCS Flow registration to determine which BCMCS Flows to transmit over the air interface.

The access network shall send a BroadcastReset message when it detects that the access terminal has entered a zone where the access terminal must refresh parameters (such as BCMCS Flow IDs, Broadcast Access Keys, etc.) related to the broadcast packet data service.

The access network may send a BroadcastReject message when it detects that one or more BCMCS Flows requested by the access terminal are not available.

If the IncludeAuthSignature field of the BroadcastParameters message is '1', then the access network may verify that the access terminal registering the BCMCS Flows has the BroadcastAccessKey required to be able to decrypt the broadcast content. The access network shall perform this verification by computing an AuthSignature using the BroadcastAccessKey corresponding to the BAKSequenceNumber included by the access terminal in the BCMCSFlowRegistration message. The access network shall compute TimeStampLong as follows:

- If RTCTimeStampShortLength is zero, then the access network shall set TimeStampLong to the TimeStampLongLength least significant bits of the system time corresponding to the time when the access terminal began transmission of the BCMCSFlowRegistration message in units of $2^{\text{TimeStampUnits}}$ Superframes.⁴⁷ Otherwise, if RTCTimeStampShortLength is not zero, the access network shall derive TimeStampLong from the TimeStampShort field included in the BCMCSFlowRegistration message using Equation (30).

$$\text{TimeStampLong} = (\text{SystemTime} - (\text{SystemTime}[\text{RTCTimeStampShortLength}-1:0] - \text{TimeStampShort}) \bmod 2^{\text{RTCTimeStampShortLength}}) \bmod 2^{\text{TimeStampLongLength}} \quad (30)$$

where

SystemTime is the current System Time in units of $2^{\text{TimeStampUnits}}$ Superframes
 TimeStampShort is a field included in the BCMCSFlowRegistration message
 SystemTime[n-1:0] represents the n least significant bits of the SystemTime

14.2.2.3.3 Access terminal requirements

14.2.2.3.3.1 BroadcastParameters message processing

The access terminals receiving broadcast content shall monitor the BroadcastParameters message. Other access terminals may ignore this message.

14.2.2.3.3.2 BCMCS flow registration

BCMCS Flow registration for paging assists the access network in determining on which ChannelBand(s) to send messages to the access terminal. BCMCS Flow registration for dynamic broadcast assists the access network in determining which BCMCS Flows should be assigned and de-assigned to appropriate physical channels.

The access terminal shall maintain a timer called RegistrationTimer with time increments of 80 ms. The access terminal shall maintain an indicator of BCMCS Flow registration enable status, called RegistrationEnabled. If PagingRegistrationPeriod is zero, and DynamicBroadcastRegistrationPeriod is zero, the access terminal shall set RegistrationEnabled to NO. Otherwise, the access terminal shall set RegistrationEnabled to YES. Whenever the access terminal initializes RegistrationEnabled to YES, or changes RegistrationEnabled from NO to YES, it shall set RegistrationTimer to a pseudorandom number (see Clause 5) between zero and $\text{Max}(2^{\text{PagingRegistrationPeriod}}, 2^{\text{DynamicBroadcastRegistrationPeriod}}) \times 80$ ms. The access terminal shall increment RegistrationTimer whenever RegistrationEnabled is YES.

The access terminal performs BCMCS Flow registration by sending a BCMCSFlowRegistration message. The access terminal shall not include in a BCMCSFlowRegistration message any BCMCS Flows that were included in any BCMCS Flow registration during the past $\text{NumPeriods} \times 2^{\text{MonitorPeriod}} \times 80$ ms interval and were not monitored by the access terminal during the past $2^{\text{MonitorPeriod}} \times 80$ ms interval.

A BCMCS Flow registration transmitted shall be considered successful when the BCMCSFlowRegistration message is transmitted on the Reverse Traffic Channel. The access terminal shall not send a BCMCSFlowRegistration message containing zero BCMCS Flows. If there are no BCMCS Flows to be registered, the BCMCS Flow registration shall be considered successful without sending a BCMCSFlowRegistration message.

⁴⁷ The access network derives the time when the access terminal began transmission of the BCMCSFlowRegistration message from the time when the BCMCSFlowRegistration message is received.

If PagingRegistrationPeriod is set to zero, the access terminal shall not perform BCMCS Flow registration for paging. Otherwise, the access terminal shall perform BCMCS Flow registration for paging as defined in 14.2.2.3.3.2.1.

If DynamicBroadcastRegistrationPeriod is set to zero, the access terminal shall not perform BCMCS Flow registration for dynamic broadcast. Otherwise, the access terminal shall perform BCMCS Flow registration for dynamic broadcast as defined in 14.2.2.3.3.2.2.

If the BCMCS Flow registration for paging procedures in 14.2.2.3.3.2.1, and the BCMCS Flow registration for dynamic broadcast procedures defined in 14.2.2.3.3.2.2 trigger a BCMCS Flow registration at the same time, then the access terminal shall combine all requested BCMCS Flows in one BCMCS Flow registration.

The access terminal may send a BCMCSFlowRegistration message even when the BCMCS Flow registration is not triggered by the BCMCS Flow registration for paging procedures in 14.2.2.3.3.2.1, or by the BCMCS Flow registration for dynamic broadcast procedures defined in 14.2.2.3.3.2.2. However, the access terminal shall not send a BCMCSFlowRegistration message containing any BCMCS Flow that was included in the last successful BCMCS Flow registration unless this BCMCS Flow registration is triggered by the BCMCS Flow registration for paging procedures in 14.2.2.3.3.2.1, or unless this BCMCS Flow registration is triggered by the BCMCS Flow registration for dynamic broadcast procedures defined in 14.2.2.3.3.2.2, or unless this BCMCS Flow registration is performed on a sector different than the last successful BCMCS Flow registration.

14.2.2.3.3.2.1 BCMCS flow registration for paging

The access terminal shall compute and store a timer expiration value (PagingRegistrationTimerMax) as

$$\text{PagingRegistrationTimerMax} = 2^{\text{PagingRegistrationPeriod}} \times 80 \text{ ms}$$

The access terminal shall perform BCMCS Flow registration when the RegistrationTimer reaches a positive integer multiple of PagingRegistrationTimerMax. The access terminal shall include in the BCMCSFlowRegistration message all BCMCS Flows for which the RegisterForPaging field of the last BroadcastParameters message is '1' and that it has monitored during the past $2^{\text{MonitorPeriod}} \times 80\text{ms}$ interval.

If PagingRegistrationPeriod is greater than or equal to DynamicBroadcastRegistrationPeriod, then the access terminal shall reset RegistrationTimer to zero after each successful BCMCS Flow registration for paging.

The access terminal may monitor the following ChannelBands for receiving transmissions from the access network:

- The ChannelBand selected by the access terminal using the hash function
- The ChannelBands associated with the BCMCS Flows included in the last successful BCMCS Flow registration.

The access terminal shall assume that the access network will not transmit to the access terminal on ChannelBands other than the above ChannelBands.

14.2.2.3.3.2.2 BCMCS Flow Registration for Dynamic Broadcast

The access terminal shall compute and store a timer expiration value (DynamicBroadcastRegistrationTimerMax) as

$$\text{DynamicBroadcastRegistrationTimerMax} = 2^{\text{DynamicBroadcastRegistrationPeriod}} \times 80 \text{ ms}$$

The access terminal shall perform BCMCS Flow registration when the RegistrationTimer reaches a positive integer multiple of DynamicBroadcastRegistrationTimerMax. The access terminal shall include in the BCMCSFlowRegistration message all BCMCS Flows for which the RegisterForDynamicBroadcast field of the last BroadcastParameters message is '1' and that it has monitored during the past $2^{\text{MonitorPeriod}} \times 80$ ms interval.

If the access terminal includes in the BCMCSFlowRegistration message a BCMCS Flow that is not being transmitted by the access network, then the access terminal should monitor the BroadcastParameters messages transmitted during the $(4 \times \text{BroadcastOverheadPeriod} \times 256)$ PHY frame interval following the transmission of the BCMCSFlowRegistration message to determine if the access network has added the requested BCMCS Flow to the BroadcastParameters message.

If DynamicBroadcastRegistrationPeriod is greater than PagingRegistrationPeriod, then the access terminal shall reset RegistrationTimer to zero after each successful BCMCS Flow registration for dynamic broadcast.

14.2.2.3.3.2.3 Computation of authorization signature

If the IncludeAuthSignature parameter of the BroadcastParameters message is '0', the BCMCS Flow included in the BCMCSFlowRegistration message does not use a BroadcastAccessKey, or the access terminal is not directed to add AuthSignature by the upper layers, then the access terminal shall omit the AuthSignature from the BCMCSFlowRegistration message if any of the following conditions is true:

- IncludeAuthSignature parameter of the BroadcastParameters message is '0'.
- The BCMCS Flow included in the BCMCSFlowRegistration message does not use a BroadcastAccessKey.
- The access terminal is not directed to add AuthSignature by the upper layers.

Otherwise, the access terminal shall include in the BCMCSFlowRegistration message an AuthSignature value computed for the BCMCS Flow included in the message. The access terminal shall compute the AuthSignature value as follows:

- The access terminal shall create an *EHMACSHA256-Message* input working buffer as shown in Table 379.

Table 379—Subfields of the EHMACSHA256-Message input working buffer

Subfield	Length (bits)
TimeStampLong	TimeStampLongLength

TimeStampLong

If RTCTimeStampShortLength is zero, then the access terminal shall set this field to the TimeStampLongLength LSBs of the System Time, in units of $2^{\text{TimeStampUnits}}$ Superframes, corresponding to the time when the access terminal will begin transmission of the BCMCSFlowRegistration message. Otherwise, the access terminal shall set this field to the TimeStampLongLength LSBs of the System Time, in units of $2^{\text{TimeStampUnits}}$ Superframes, corresponding to a time that is not later than when the access terminal will begin transmission of the BCMCSFlowRegistration message that will carry this AuthSignature.⁴⁸ The access terminal shall use the same TimeStampLong field for computing the AuthSignature parameters for all BCMCS Flows included in a BCMCSFlowRegistration message.

⁴⁸For example, the access terminal may choose the current CDMA System time as TimeStampLong.

- The access terminal shall perform the EHMACHA256 procedure as specified in 3GPP2 S.S0078 with the following input parameters:
 - The KEY parameter of EHMACHA256 shall be set to the Broadcast Access Key.
 - The L_KEY parameter of EHMACHA256 shall be set to 16, identifying the length of the Broadcast Access Key in octets.
 - The MESSAGE parameter of EHMACHA256 shall be set to the pointer to the most significant bit of the EHMACHA256 Message input working buffer.
 - The MESSAGE_OFFSET parameter of EHMACHA256 shall be set to 0.
 - The MESSAGE_LENGTH parameter of EHMACHA256 shall be set to the number of bits of data in the EHMACHA256 Message input working buffer.
 - The HMAC parameter of EHMACHA256 shall be set to the pointer to the most significant bit of the output buffer that will contain the computed MAC value.
 - The L_HMAC parameter of EHMACHA256 shall be set to 4, identifying the length of the output in octets.
- The access terminal shall set the AuthSignature to the contents of the output buffer indicated by HMAC.

14.2.2.3.4 Message formats

14.2.2.3.4.1 BroadcastParameters message

The access network sends the BroadcastParameters message to provide the access terminal with the mapping between logical channels and Broadcast Physical Channels (see Table 380).

Table 380—BroadcastParameters message

Field	Length (bits)
MessageID	8
ProtocolSubtype	16
BroadcastParametersSignature	16
CompleteListInd	1
QCISignature	20
AllReservedInterlaces	1
BCMCSReservedInterlaces	0 or 4
RotationalAngleIncluded	1
RotationalAngle	0 or 4
NumBOC	2

NumBOC occurrences of the following field:

BCMCSFlowIDLength	2
BCMCSOverheadFields	Variable Length

Padding	0–7 (as needed)

MessageID	The access network shall set this field to 0x00.
ProtocolSubtype	The access network shall set this field to the Protocol Subtype corresponding to this BCMCS Protocol Suite as specified in 15.2.4.
BroadcastParametersSignature	The access network shall increment this field if any of the other fields in the BroadcastParameters message changes.
CompleteListInd	The access network shall set this field to '1' if this message includes all the available BCMC flows; otherwise, the access network shall set this field to '0'.
QCISignature	The access network shall set this field to the QCISignature public data of the Overhead Messages protocol.
AllReservedInterlaces	The access network shall set this field to '1' to indicate that all the subbands of all the reserved interlaces are being used for BCMCS; otherwise, the access network shall set this field to '0'.
BCMCSReservedInterlaces	If the AllReservedInterlaces field is set to '1', then the access network shall omit this field; otherwise, the access network shall include this field and set it according to Table 381. All the subbands in these interlaces shall be used for BCMCS.

Table 381—Interpretation of BCMCSReservedInterlaces

Value	Reserved BCMCS interlaces
0000	0
0001	0, 1
0010	0, 1, 2
0011	0, 1, 2, 3
0100	0, 1, 2, 3, 4
0101	0, 1, 2, 3, 4, 5
0110	0, 1, 2, 3, 4, 5, 6
0111	3
1000	6
1001	0, 2
1010	0, 4
1011	0, 6
1100	2, 4
1101	2, 6
1110	4, 6
1111	2, 4, 6

RotationalAngleIncluded The access network shall set this field to '0' if all ModulationLayer fields in the BCMCSOverheadFields are set to '0'; otherwise, the access network shall set this field to '1'.

RotationalAngle If the RotationalAngleIncluded field is set to '0', the access network shall omit this field; otherwise, the access network shall include this field and set it as follows: the access network shall set this field to indicate the initial reference phase for the enhancement-layer signal constellation of the enhanced hierarchical modulation in units of degree. The procedure of the enhanced hierarchical modulation is detailed in 9.2.6.7.5. The value of this parameter shall be interpreted as specified in to Table 382.

Table 382—Interpretation of RotationalAngle

Field value	Interpretation
	Unit: degree
0000	0.0
0001	2.81
0011	5.63
0010	8.44
0110	11.25
0111	14.06
0101	16.88
0100	19.69
1100	22.50
1101	25.31
1111	28.13
1110	30.94
1010	33.75
1011	36.56
1001	39.38
1000	42.19

NumBOC The access network shall set this field to indicate the number of different Broadcast Overhead Channels (and hence Physical Channel groups) present in each BCMCS ultraframe.

BCMCSFlowIDLength The access network shall set this field to one less than the length of the BCMCS Flow Identifier in units of octets. The access network shall not set this field to '00'.

BCMCSOverheadFields The access network shall set this as defined in 14.2.2.3.4.1.1. This field specifies the logical to physical channel mapping for the logical channels transmitted on the Forward Broadcast and Multicast Services Channel.

Padding The access network shall add Padding bits in order to make the length of the entire message equal to an integer number of octets. The access network shall set these bits to '0'. The access terminal shall ignore these bits.

14.2.2.3.4.1.1 BCMCSOverheadFields

Table 383—BCMCSOverheadFields

BCMCSFlowCount	8
PagingRegistrationPeriod	8
DynamicBroadcastRegistrationPeriod	8
MonitorPeriod	8
NumPeriods	8
BroadcastChannelSaltLength	8
BroadcastChannelSalt	$\text{BroadcastChannelSaltLength} \times 8$
EncryptionKeyEntropy	8
IncludeAuthSignature	1
RTCTimeStampShortLength	0 or 8
TimeStampLongLength	0 or 8
TimeStampUnit	0 or 4
NumOuterFrames	2
NumOffsets	4
BOCTransmissionFormat	3
BOCPDRParametersIncluded	0 or 1
BOCPilotToDataRatioRecord	0 or Variable
FDSSeedNumMSBs	0 or 4
FDSSeedMSBs	0 or FDSSeedNumMSBs

NumOffsets occurrences of the following fields:

Offset _j	4
Period _j	4

BCMCSFlowCount occurrences of the following variable-length record:

BCMCSFlowID	$(\text{BCMCSFlowIDLength} + 1) \times 8$
RegisterForPaging	1
RegisterForDynamicBroadcast	1

Table 383—BCMCSOverheadFields (continued)

LogicalChannelSameAsPreviousBCMCSFlow	1
NumGuardSubcarriers	6
StartLocation	0 or 8
Duration	0 or 4
BCMCSTransmissionFormat	0 or 6
OuterCode	0 or 4
Period	0 or 3
PilotStaggerIndex	0 or 3
BCMCSPIlotToDataRatioRecord	0 or Variable
FDSSeedSameAsPreviousLogCh	0 or 1
FDSSeedLSBs	0 or 10 – FDSSeedNumMSBs
ModulationLayer	0 or 1

BCMCSFlowCount The access network shall set this field to the number of BCMCS Flow identifiers included in the BCMCSOverheadFields record of this message.

PagingRegistrationPeriod The access network shall set this field such that the desired timer value for timer-based BCMCS Flow registration for paging is $2^{\text{PagingRegistrationPeriod}} \times 80$ ms.

The access network shall not set this field to a value larger than 0x15.

DynamicBroadcastRegistrationPeriod

The access network shall set this field such that the desired timer value for Timer-based BCMCS Flow registration for dynamic assignment of BCMCS Flows to physical channels is $2^{\text{PagingRegistrationPeriod}} \times 80$ ms.

The access network shall not set this field to a value larger than 0x15.

MonitorPeriod The access network shall set this field such that the desired timer value is $2^{\text{MonitorPeriod}} \times 80$ ms.

The access network shall not set this field to a value larger than 0x0f.

NumPeriods The access network shall set this field to the desired value for NumPeriods.

The access network shall not set this field to a value larger than 0x04.

BroadcastChannelSaltLength

The access network shall set this field to the length of the BroadcastChannelSalt field in octets.

BroadcastChannelSalt The access network shall set this field to the value of the *Salt* input parameter that is to be used in the KeyStrengthRedAlg procedure specified in 3GPP2 S.S0078 for generating Short Term Keys.

- EncryptionKeyEntropy** The access network shall set this field to the value of the KeyEntropy parameter that is to be used in the KeyStrengthRedAlg procedure specified in 3GPP2 S.S0078 for generating Short Term Keys. Valid values for this field are 0 through 16 inclusive.
- IncludeAuthSignature** The access network shall set this field to '1' if the access terminal is required to include authorization signature for BCMCS Flows registered in a BCMCSFlowRegistration message. Otherwise, the access network shall set this field to '0'.
- RTCTimeStampShortLength** If IncludeAuthSignature is '0', the access network shall omit this field. Otherwise, the access network shall set this field to the length in units of bits of the TimeStampShort field to be included in a BCMCSFlowRegistration message, which will be transmitted on the Reverse Traffic Channel.
- TimeStampLongLength** If IncludeAuthSignature is '0', the access network shall omit this field. Otherwise, the access network shall set this field to the length in units of bits of the TimeStampLong field to be used in the computation of the AuthSignature included in a BCMCSFlowRegistration message.
- TimeStampUnit** If IncludeAuthSignature is '0', the access network shall omit this field. Otherwise, the access network shall set this field to specify the unit of the TimeStampLong field. The unit of the TimeStampLong field in slots shall be 2 to the power of the value of this field.
- NumOuterFrames** The access network shall set this field to the number of outerframes per ultraframe as indicated in Table 384.

Table 384—Description of NumOuterFrames field

NumOuterFrames field	Number of outerframes per ultraframe
'00'	1
'01'	2
'10'	4
'11'	8

- NumOffsets** The access network shall set this field to the number of offsets that the assigned to the outerframes as described in 14.2.6.3.1.3.
- BOCTransmissionFormat** The access network shall set this field to the packet format index of the Broadcast Overhead Channel as specified in 14.2.6.3.1.6.4.
- BOCPDRParametersIncluded** If the BCMCSFlowCount field is set to zero, the access network shall omit this field. Otherwise, the access network shall include this field and set it as follows:

If the pilot to data ratio of the Broadcast Overhead Channel is not the same as PDRBOCDefault, then the access network shall set this field to '1'. Otherwise, the access network shall set this field to '0'.

BOCPilotToDataRatioRecord

If the BOCPDRParametersIncluded field is set to '0', then the access network shall omit this field. Otherwise, the access network shall include this field and set it according to 14.2.2.3.4.1.2.

FDSSeedNumMSBs

If BCMCSFlowCount field is set to zero, the access network shall omit this field. Otherwise, the access network shall include this field and set it to the unsigned binary representation of a value between 0 to 10, inclusive, to indicate the length of the FDSSeedMSBs field. FDSSeedNumMSBs shall be such that the FDSSeedNumMSBs MSBs of the 10-bit FDSSeed parameters of all logical channels defined after this field for which NumOffsets is greater than zero shall be the same.

FDSSeedMSBs

If BCMCSFlowCount field is set to zero, the access network shall omit this field. Otherwise, the access network shall include this field and set it to the common FDSSeedNumMSBs MSBs of the FDSSeed parameters of all logical channels defined after this field for which the NumOffsets is greater than zero.

Offset_j

The access network shall set this field to the value offset_j – 1 as described in 14.2.6.3.1.3.

Period_j

The access network shall set this field to the value period_j – 1 as described in 14.2.6.3.1.3.

BCMCSFlowID

The access network shall set this field to the BCMCS Flow identifier of this BCMCS Flow.

RegisterForPaging

The access network shall set this field to '1' if the access terminal is required to include this BCMCS Flow in a BCMCSFlowRegistration message to allow the access network to send messages to the access terminal on the appropriate channel(s). Otherwise, the access network shall set this field to '0'.

RegisterForDynamicBroadcast

The access network shall set this field to '1' if the access terminal is required to include this BCMCS Flow in a BCMCSFlowRegistration message to allow the access network to dynamically assign and de-assign BCMCS Flows to physical channels. Otherwise, the access network shall set this field to '0'.

LogicalChannelSameAsPreviousBCMCSFlow

If this BCMCS Flow is transmitted using the same logical channel as the previous BCMCS Flow listed in the BCMCSOverheadFields field of this message, then the access network shall set this field to '1'. Otherwise, the access network shall set this field to '0'. If this is the first BCMCS Flow listed in the BCMCSOverheadFields field of this message, then the access network shall set this field to '0'.

Records for all BCMCS

Flow IDs that share the same logical channel shall be placed consecutively in the BroadcastParameters message.

NumGuardSubcarriers

This field shall be set to the number of GuardSubcarriers, in units of 2.

Start Location If LogicalChannelSameAsPreviousBCMCSFlow is set to '1', then the access network shall omit this field; otherwise, the access network shall include this field and set this field to the starting location of the logical channel in number of Subbands_i as specified in 14.2.6.3.1.3.

Duration If LogicalChannelSameAsPreviousBCMCSFlow is set to '1', then the access network shall omit this field; otherwise, the access network shall include this field and set this field to the number of number of consecutive BCMCS subbands that the logical channel occupies as specified in Table 406.

BCMCSTransmissionFormat If LogicalChannelSameAsPreviousBCMCSFlow is set to '1', then the access network shall omit this field. Otherwise, the access network shall set this field to the TransmissionFormat parameter of this logical channel according to Table 410 to indicate the Broadcast transmission format of this logical channel.

OuterCode If LogicalChannelSameAsPreviousBCMCSFlow is set to '1', or if NumOffsets is zero, then the access network shall omit this field. Otherwise, the access network shall set this field to indicate which Reed-Solomon outer code is used to form an error control block for this logical channel according to Table 385.

Table 385—Description of OuterCode field

OuterCode	(N, K, R) Reed-Solomon outer code
'0000'	(1, 1, 0) Reed-Solomon code (No outer code)
'0001'	(16, 12, 4) Reed-Solomon code
'0010'	(16, 13, 3) Reed-Solomon code
'0011'	(16, 14, 2) Reed-Solomon code
'0100'	(32, 24, 8) Reed-Solomon code
'0101'	(32, 26, 6) Reed-Solomon code
'0110'	(32, 28, 4) Reed-Solomon code
All other values are reserved	

Period If LogicalChannelSameAsPreviousBCMCSFlow field is set to '1', then the access network shall omit this field. Otherwise, the access network shall set this field to the 2-bit representation of the Period parameter associated with this logical channel as shown in Table 386.

Table 386—Description of Period field

Period field	Number of transmissions
'000'	1
'001'	2
'010'	3
'011'	Reserved

PilotStaggerIndex If LogicalChannelSameAsPreviousBCMCSFlow is set to '1', or if NumOffsets is zero, then the access network shall omit this field. Otherwise, the access network shall set this field to indicate the content dependent PilotStagger parameter according to Table 387. For different contents broadcast at the same time the access network should assign different PilotStaggerIndex.

Table 387—Description of PilotStaggerIndex field

PilotStaggerIndex	PilotStagger
'000'	0
'001'	1
'010'	2
'011'	3
'100'	4
'101'	5
'110'	6
'111'	7

BCMCS PilotToDataRatioRecord

If LogicalChannelSameAsPreviousBCMCSFlow is set to '1', or if NumOffsets is zero, or if the BOC PDRParametersIncluded field is set to '0', then the access network shall omit this field. Otherwise, the access network shall include this field and set it according to 14.2.2.3.4.1.2.

FDSSeedSameAsPreviousLogCh

If LogicalChannelSameAsPreviousBCMCSFlow is set to '1', or if NumOffsets is zero, or if $(10 - \text{FDSSeedNumMSBs})$ is zero, the access network shall omit this field. Otherwise, the access network shall include this field and set it as follows:

If there exists an occurrence of a non-empty FDSSeedLSBs field prior to this field and the last such occurrence is set equal to the $(10 - \text{FDSSeedMSBs})$ LSBs of the FDSSeed parameter of this logical channel, then the access network shall set this field to '1'. Otherwise, the access network shall set this field to '0'.

FDSSeedLSBs

If the FDSSeedSameAsPreviousLogCh is included and set to '0', the access network shall include this field and set it to the $(10 - \text{FDSSeedNumMSBs})$ LSBs of the FDSSeed parameter for this logical channel. Otherwise, the access network shall omit this field.

ModulationLayer

If the logical channel is transmitted on the base layer, the access network shall set this field to '0'. If the logical channel is transmitted on the enhancement layer, the access network shall set this field to '1'.

14.2.2.3.4.1.2 BCMCSPilotToDataRatioRecord

Table 388—BCMCSPilotToDataRatioRecord

Field	Length (bits)
BCMCS PDR Provided For This Logical Ch	1
BCMCS PDR Same As Before	0 or 1
BCMCS Pilot To Data Ratio	0 or 4
Pilot To Data Ratio	0 or 4

BCMCS PDR Provided For This Logical Ch
The access network shall set this field to '1' if the pilot to data ratio of the Broadcast packet is different from the PDR BOC Default.

BCMCS PDR Same As Before
If the BCMCS PDR Record For This Logical Ch field is set to '0', the access network shall omit this field. Otherwise, the access network shall include this field and set it as follows:

If the BCMCS Overhead Fields field of the Broadcast Parameters message specifies at least one logical channel prior to this logical channel that includes a non-empty BCMCS PDR Record For This Logical Ch field and has the same BCMCS Transmission Format as this logical channel, and if the last one of such logical channels has the same values for all in-use pilot-to-data-ratio parameters described in 14.2.6.3.1.6.6.1 as this logical channel, then the access network shall set this field to '1'. Otherwise, the access network shall set this field to '0'.

BCMCS Pilot To Data Ratio
The access network shall set this field as described in 14.2.6.3.1.6.6.1. The method by which the access network sets the BCMCS Pilot To Data Ratio parameter is beyond the scope of this standard.

Pilot To Data Ratio
The access network shall set this field as described in 14.2.6.3.1.6.6.1. The method by which the access network sets the Pilot To Data Ratio parameter is beyond the scope of this standard.

Channels	FTC	RLP	Best effort
Addressing	Broadcast	Auth Tag	Not applicable

14.2.2.3.4.2 BCMCSFlowRegistration message

The access terminal sends the BCMCSFlowRegistration message to notify the access network of the list of BCMCS Flows that the access terminal intends to monitor (see Table 389).

Table 389—BCMCSFlowRegistration message

Field	Length (bits)
MessageID	8

Table 389—BCMCSFlowRegistration message (continued)

Field	Length (bits)
TimeStampShortIncluded	1
TimeStampShortLength	0 or 8
TimeStampShort	0 or TimeStampShortLength
BCMCSFlowIDLength	2

Zero or one occurrence of the following record:

BCMCSFlowCount	6

BCMCSFlowCount occurrences of the following four fields:

BCMCSFlowID	$(\text{BCMCSFlowIDLength} + 1) \times 8$
AuthSignatureIncluded	1
BAKSequenceNumber	0 or 4
AuthSignature	0 or 32

Padding	0 – 7 (as needed)

- MessageID** The access terminal shall set this field to 0x01.
- TimeStampShortIncluded** The access terminal shall set this field to '1' if an AuthSignature, AuthSignature1, or AuthSignature2 field is included in this message.
- TimeStampShortLength** If TimeStampShortIncluded is '0', the access terminal shall omit this field. If TimeStampShortIncluded is '1', then the access terminal shall set this field to the value of the RTCTimeStampShortLengthfield in the BroadcastParameters message.
- TimeStampShort** If TimeStampShortIncluded is '0', the access terminal shall omit this field. Otherwise, the access terminal shall set this field to the RTCTimeStampShortLength least significant bits of the TimeStampLong parameter used to generate the AuthSignature parameters included in this message.
- BCMCSFlowIDLength** The access terminal shall set this field to the BCMCSFlowIDLength received in the BroadcastParameters message.
- BCMCSFlowCount** The access terminal shall set this field to the number of BCMCS Flow identifiers that follow this field in the message. The access terminal shall not set this field to zero.
- BCMCSFlowID** The access terminal shall set this field to the identifier of the BCMCS Flow.
- AuthSignatureIncluded** The access terminal shall set this field as follows:
- The access terminal shall set this field to '1' if authorization signature is included for this BCMCS Flow. Otherwise, the access terminal shall set this field to '0'.

BAKSequenceNumber	If the AuthSignatureIncluded field is '0', the access terminal shall omit this field. Otherwise, the access terminal shall set this field to the sequence number of the BroadcastAccessKey that is used to generate the AuthSignature for this BCMCS Flow.
AuthSignature	If AuthSignatureIncluded is '0', the access terminal shall omit this field. Otherwise, the access terminal shall set this field to the AuthSignature value that is computed as described in 14.2.2.3.3.2.3.
Padding	The access terminal shall add Padding bits in order to make the length of the entire message equal to an integer number of octets. The access terminal shall set these bits to '0'. The access terminal shall ignore these bits.

The access terminal shall use the following values for the message information fields when transmitting the BCMCSFlowRegistration message:

Channels	RTC	RLP	Reliable
Addressing	Unicast	AuthTag	Not applicable

14.2.2.3.4.3 BroadcastReject message

The access network can send a BroadcastReject message to inform the access terminal that one or more BCMCS Flows requested by the user are rejected (see Table 390).

Table 390—BroadcastReject message

Field	Length
MessageID	8
BCMCSFlowIDLength	2
Zero or one occurrence of the following record:	
BCMCSFlowCount	6
BCMCSFlowCount occurrences of the following three fields:	
BCMCSFlowID	$(\text{BCMCSFlowIDLength}+1) \times 8$
SameAsPreviousBCMCSFlow	1
RejectReason	0 or 4
Padding	0–7 (if needed)

MessageID	The access network shall set this field to 0x02.
BCMCSFlowIDLength	The access network shall set this field to the BCMCSFlowIDLength field of the BroadcastParameters message.
BCMCSFlowCount	The access network shall set this field to the number of BCMCS Flow identifiers included in this message. The access network shall not set this field to zero.

BCMCSFlowID The access network shall set this field to the BCMCS Flow identifier of the BCMCS Flow.

SameAsPreviousBCMCSFlow

If this is the first BCMCS Flow in this message, the access network shall set this field to '0'. If the RejectReason field is the same for this BCMCS Flow as for the previous BCMCS Flow in this message, then the access network shall set this field to '1'. Otherwise, the access network shall set this field to '0'.

RejectReason

If the SameAsPreviousBCMCSFlow field is set to '1', then the access network shall omit this field. Otherwise, the access network shall include this field and shall set it as specified in Table 391.

Table 391—Reject reason values

Value	RejectReason
'0000'	BCMCS Flow ID not available
'0001'	BCMCS Flow ID not transmitted
'0010'	Invalid authorization signature
'0011'–'1111'	Reserved

Padding

The access network shall add Padding bits in order to make the length of the entire message equal to an integer number of octets. The access network shall set these bits to zero. The access terminal shall ignore these bits.

The access network shall use the following values for the message information fields when transmitting the BroadcastReject message:

Channels	FTC

RLP	Best effort

Addressing	Broadcast	Unicast

AuthTag	Not applicable

14.2.2.3.4.4 BroadcastReset message

The access network sends the BroadcastReset message to inform the access terminal that it must refresh parameters (such as BCMCS Flow ID, Broadcast Access Key, etc.) for BCMCS Flows it intends to monitor (see Table 392).

Table 392—BroadcastReset message

Field	Length
MessageID	8

MessageID

The access network shall set this field to 0x03.

The access network shall use the following values for the message information fields when transmitting the BroadcastReset message:

Channels	FTC

RLP	Best effort

Addressing	Unicast

AuthTag	Not applicable

14.2.2.3.5 Interfaces to other protocols

14.2.2.3.5.1 Indications

This protocol registers to receive the following indications:

14.2.3 Broadcast Packet Consolidation Protocol

14.2.3.1 Overview

The Broadcast Packet Consolidation Protocol performs the following functions:

- Framing of higher-layer packets
- Multiplexing of Broadcast content packets, Broadcast signaling messages, and tunneled unicast Route packets

14.2.3.2 Protocol Data Unit

The transmission unit of this protocol is a Broadcast Packet Consolidation Protocol Packet. In the transmit direction, the Broadcast Packet Consolidation Protocol fragments the higher-layer packet, if necessary, and forms Broadcast Packet Consolidation Protocol packets. A Broadcast Packet Consolidation Protocol packet contains a one octet or two-octet long header.

Figure 164 shows the Broadcast Packet Consolidation Protocol encapsulation for the 2-octet PCP header and Figure 165 shows the Broadcast Packet Consolidation Protocol encapsulation for the 1-octet PCP header.

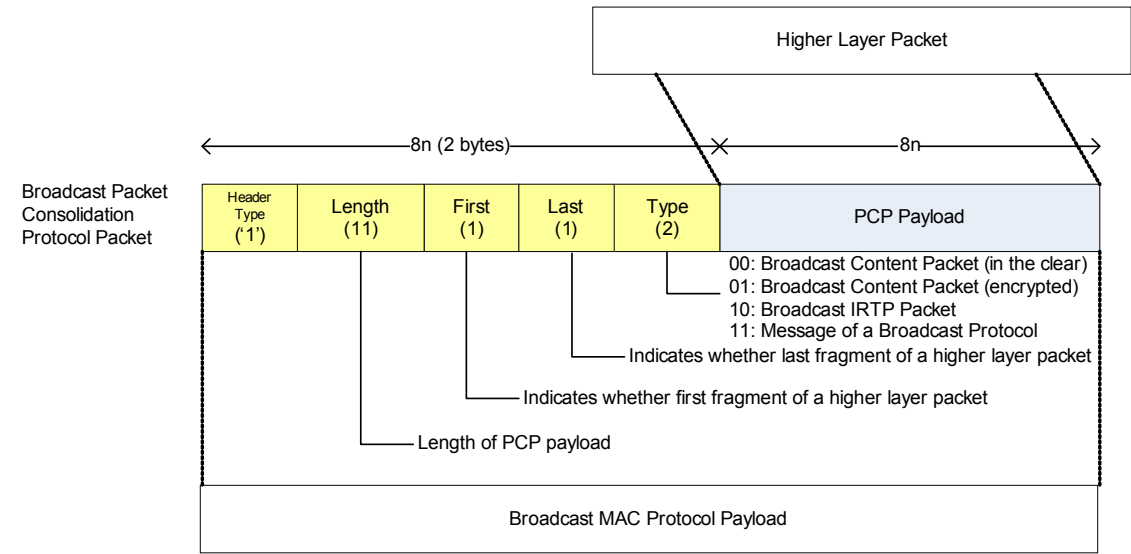


Figure 164—Broadcast Packet Consolidation Protocol Encapsulation for 2-octet header

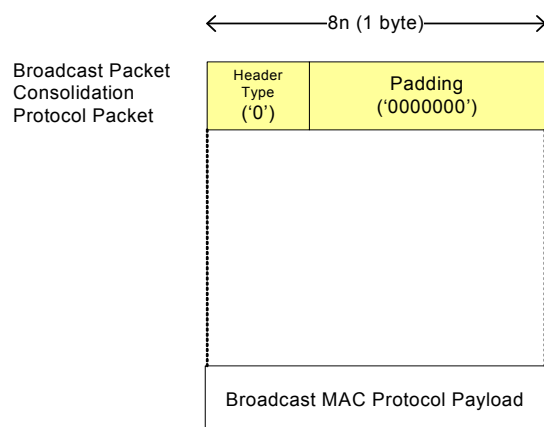


Figure 165—Broadcast Packet Consolidation Protocol Encapsulation for 1-octet header

14.2.3.3 Procedures

14.2.3.3.1 Access network requirements

The access network shall fragment the higher-layer packets if necessary, to create a Broadcast Packet Consolidation Protocol packet payload.

The access network shall add the Broadcast Packet Consolidation Protocol header defined in 14.2.3.3.2 in front of every Broadcast Packet Consolidation Protocol packet payload to create a Broadcast Packet Consolidation Protocol packet.

The length of the resulting Broadcast Packet Consolidation Protocol packet shall not exceed the length of the payload available in the Broadcast Lower MAC Sublayer packet that will carry the Broadcast Packet Consolidation Protocol packet.

If the PayloadType field of the Broadcast Packet Consolidation Protocol packet is set to '01', the access network shall invoke the Broadcast Security Protocol to encrypt the payload of this Broadcast Packet Consolidation Protocol packet.

The access network shall forward the Broadcast Packet Consolidation Protocol packet for transmission to the Broadcast MAC protocol.

14.2.3.3.2 Broadcast Packet Consolidation Protocol header format

The access network shall place the following header at the beginning of every fragment of a higher-layer packet (see Table 393).

Table 393—Broadcast Packet Consolidation Protocol header format

Field	Length (bits)
HeaderType	1
Padding	7 ^a
Length	11 ^a

Table 393—Broadcast Packet Consolidation Protocol header format (continued)

Field	Length (bits)
First	1 ^a
Last	1 ^a
PayloadType	2 ^a

^aThis field may be absent (see field description for conditions).

HeaderType	The access network shall set this field to '0' if the Header is 1-octet long; the access network shall set this field to '1' if the Header is 2-octets long
Padding	The access network shall omit this field if the HeaderType field is set to '1'; otherwise, the access network shall include this field and set it to '0000000'.
Length	The access network shall omit this field if the HeaderType field is set to '0'; otherwise, the access network shall include this field and set this field to the number of octets in the payload field of this Broadcast Packet Consolidation Protocol packet. If this is a fill packet, the length shall be set to zero.
First	The access network shall omit this field if the HeaderType field is set to '0'; otherwise, the access network shall include this field and set it as follows: If the payload field of this Broadcast Packet Consolidation Protocol packet is the first fragment of a higher-layer packet or the Length field is set to zero, then the access network shall set this field to '1'. Otherwise, the access network shall set this field to '0'.
Last	The access network shall omit this field if the HeaderType field is set to '0'; otherwise, the access network shall include this field and set it as follows: If the payload field of this Broadcast Packet Consolidation Protocol packet is the last fragment of a higher-layer packet or the Length field is set to zero, then the access network shall set this field to '1'. Otherwise, the access network shall set this field to '0'.
PayloadType	The access network shall omit this field if the HeaderType field is set to '0'; otherwise, the access network shall include this field and set it as follows: If the Length field is set to zero, then the access network shall set this field to '00'; otherwise, the access network shall set this field as specified in Table 394 to indicate the type of payload carried in this Broadcast Packet Consolidation Protocol packet.

Table 394—PayloadType

Value	PayloadType
'00'	BCMC content (un-encrypted)
'01'	BCMC content (encrypted)
'10'	Broadcast Inter-Route Tunneling Protocol Packet
'11'	'Signaling message from a protocol in the Broadcast Protocol Suite

14.2.4 Broadcast Security Protocol

The Broadcast Security Protocol provides the procedures for encrypting Broadcast Packet Consolidation Protocol packet payload of PayloadType '01' (i.e., contains BCMC content that is to be encrypted). If such a payload is to be encrypted, the Broadcast Security protocol generates a Short Term Key and a cryptosync, and uses the Short Term Key, the cryptosync, and other parameters (if any) to encrypt the payload. The access network encrypts the payload using the AES (also known as Rijndael) encryption procedures defined in 3GPP2 S.S0078.

14.2.4.1 Access network requirements

The access network shall transmit Security Parameters messages using requirements specified in 14.2.4.1.1.5.

If the Broadcast Packet Consolidation Protocol packet payload is to be encrypted, the access network shall perform the following:

- If this is the first Broadcast Packet Consolidation Protocol packet in an error control block (see Clause 4 for description of error control block), the access network shall generate an encryption mask for the error control block as described in 14.2.4.1.1.
- The access network shall encrypt the Broadcast Packet Consolidation Protocol packet payload by exclusive-ORing the corresponding bits of the encryption mask with the Broadcast Packet Consolidation Protocol packet payload as described in 14.2.4.

14.2.4.1.1 Constructing the encryption mask

The access network shall generate the encryption mask as follows:

- The access network shall generate a RandomSeed as described in 14.2.4.1.1.1 for encrypting the error control block.
- The access network shall generate the Short Term Key for the error control block as described in 14.2.4.1.1.2.
- The access network shall generate a cryptosync for the error control block as described in 14.2.4.1.1.3.
- The access network shall generate the encryption mask for the error control block by calling the ESP_AES procedure specified in 3GPP2 S.S0078 with its inputs set as follows:
 - Set the *key* to the Short Term Key for the error control block under consideration.
 - Set *fresh* to the value of the cryptosync for the error control block under consideration.
 - Set the *freshsize* to 8.
 - Set the *buf* to the address of the beginning of a memory space that contains $(K \times \text{MACPacketSize} \times 8)$ bits, all set to zero, where K and MACPacketSize are parameters of the error control block for the logical channel under consideration (see Clause 4).
 - Set the *bit_offset* to zero.
 - Set the *bit_count* to $(K \times \text{MACPacketSize} \times 8)$ bits.
- After the ESP_AES procedure is returned, the access network shall set the encryption mask to the output of the ESP_AES procedure, which starts at the memory space specified by *buf* and of size $(K \times \text{MACPacketSize} \times 8)$ bits.

14.2.4.1.1.1 Generating the RandomSeed

If this error control block is to be encrypted using the same Short Term Key as the previous error control block belonging to the logical channel under consideration, then the access network shall not generate a RandomSeed. Otherwise, the access network shall generate the RandomSeed by invoking the algorithmic

function $f0$ as specified in 3GPP2 S.S0078. The access network shall set the input parameters of $f0$ as follows:

- Set the K parameter to a randomly chosen seed.
- Set the fi parameter to 0x41.
- Set the Fmk parameter to 0x4243474B.

The access network shall set RandomSeed to the least significant 32 bits of the 64-bit output of $f0$.

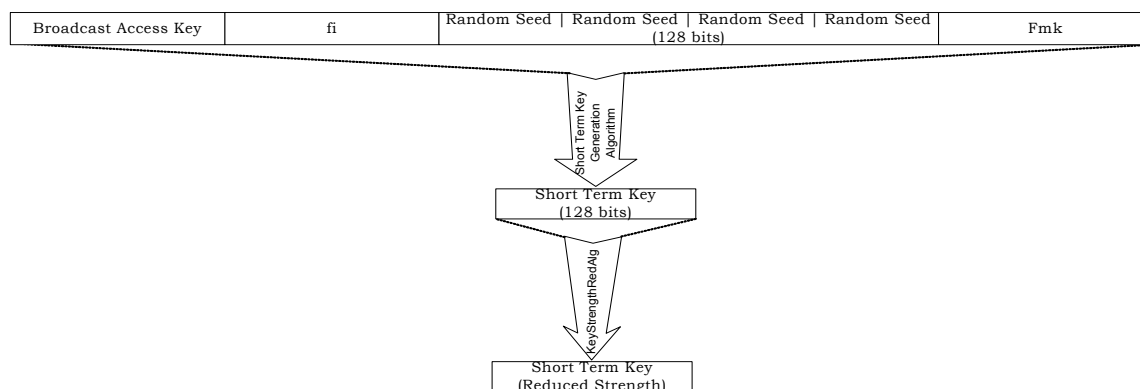
14.2.4.1.1.2 Constructing the Short Term Key

If this error control block is to be encrypted using the same Short Term Key as the previous error control block belonging to the logical channel under consideration, then the access network shall set the Short Term Key to the Short Term Key corresponding to the previous error control block. Otherwise, the access network shall generate the Short Term Key as follows:

- The access network shall invoke the algorithmic function $f3$ as specified in 3GPP2 S.S0078. The access network shall set the input parameters of $f3$ as follows:
 - Set the K parameter to the Broadcast Access Key for the logical channel under consideration.
 - Set the fi parameter to 0x45.
 - Set the RAND parameter to RandomSeed|RandomSeed|RandomSeed|RandomSeed.
 - Set the Fmk parameter to 0x4243474B.
- The access network shall set the Short Term Key to the 128-bit output of $f3$.
- If key strength reduction is required, the access network shall call the KeyStrengthRedAlg procedure specified in 3GPP2 S.S0078 with its inputs set as follows:
 - Set the KeyLength to 16.
 - Set the OriginalKey to the value of the Short Term Key.
 - Set SaltLength to the value of the BroadcastChannelSaltLength.⁴⁹
 - Set Salt to the value of the BroadcastChannelSalt2313H57.
 - Set the KeyEntropy to the value of the EncryptionKeyEntropy parameter.
- When the KeyStrengthRedAlg returns, the access network shall set the Short Term Key to RedStrengthKey, which is the output of the KeyStrengthRedAlg procedure.

The steps for generating the Short Term Key are shown in Figure 166.

⁴⁹BroadcastChannelSaltLength and BroadcastChannelSalt may be specified by the BroadcastParameters message or may be obtained by other means that are outside the scope of this standard.

**Figure 166—Generation of Short Term Key**

The access network shall generate the Short Term Key for BlocksPerRandomSeed error control blocks associated with the logical channel under consideration using each value of RandomSeed. The access network shall change the RandomSeed used to generate the Short Term Key every BlocksPerRandomSeed error control blocks associated with the logical channel under consideration. The access network shall assume that the first error control block began transmission during the first occurrence at or after time $T = 0$ of an interlace-multiplex pair assigned to the logical channel under consideration, where T is the System Time in PHY frames as defined in 5. The access network shall set the BlocksPerRandomSeed parameter to a value between 1 and 255 inclusive.

14.2.4.1.1.3 Constructing the Cryptosync

The access network shall compute the cryptosync for the Forward Broadcast Multicast Services Channel as shown in Table 395.

Table 395—Subfields of the Cryptosync

Subfield	Length (bits)
ChannelID	8
TimeStamp	56

ChannelID The access network shall set this field to 0x04 to indicate that the cryptosync is used for the Forward Broadcast Multicast Services Channel.

TimeStamp The access network shall set this field to the 56 least significant bits of the System Time in units of PHY frames, corresponding to the beginning of transmission of the first Physical Layer packet of the error control block carrying this Broadcast Security Packet.

14.2.4.1.1.4 Encryption procedures

The access network shall encrypt the Broadcast Packet Consolidation Protocol packet payload by exclusive-ORing the payload with the corresponding portion of the encryption mask. The size of the encryption mask is $K \times \text{MACPacketSize} \times 8$ bits. However, the sum of the lengths of all payloads in an error control block will be less than the size of the encryption mask, because there may be one or more payloads of PayloadType '00', '10', or '11' (i.e., not to be encrypted) and the two-octet header of each Broadcast Packet Consolidation Protocol packet is not encrypted. The access network shall discard bits of the encryption mask

corresponding to those bits of the Broadcast Packet Consolidation Protocol packet that is not to be encrypted. The access network shall then encrypt the Broadcast Packet Consolidation Protocol packet payload by exclusive-ORing the Broadcast Packet Consolidation Protocol packets with bits from the encryption mask in the order from the beginning of the encryption mask to the end. Figure 167 shows this procedure.

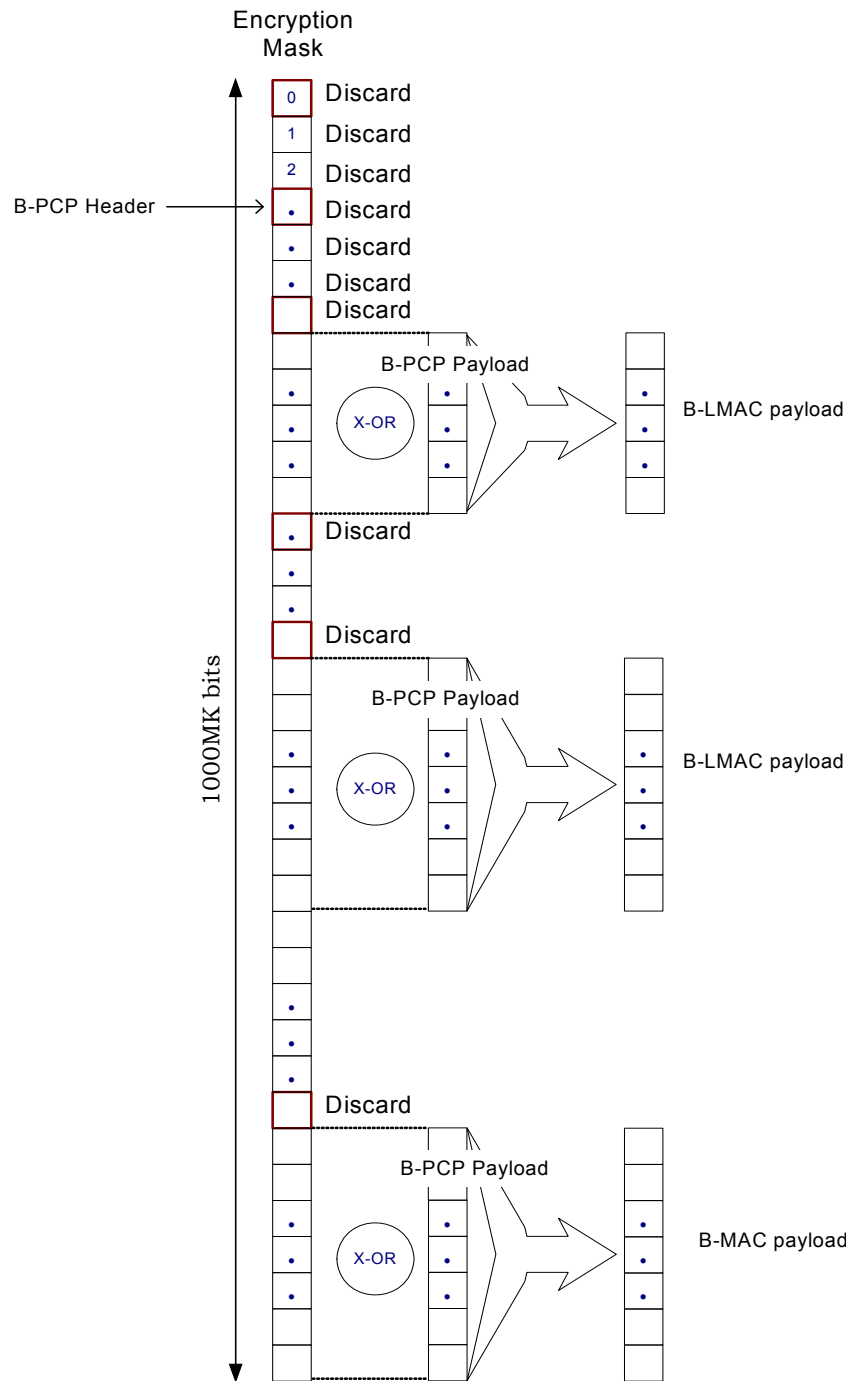


Figure 167—Generation of Broadcast Security Packets

14.2.4.1.1.5 Requirements for transmission of SecurityParameters message

If the encryption mask for an error control block is generated using a different RandomSeed than the previous error control block associated with the logical channel under consideration, then the access network shall include a SecurityParameters message in the error control block. Otherwise, the access network need not include a SecurityParameters message in the error control block.

The SecurityParameters message contains one or more occurrences of the RandomSeed field, which conveys the value of a RandomSeed. If the SecurityParameters message contains multiple occurrences of the RandomSeed field, the access network shall use the first RandomSeed to generate the encryption mask for the error control block in which this SecurityParameters message is transmitted. The access network shall use successive RandomSeeds to generate encryption masks for successive error control blocks associated with the logical channel under consideration.

14.2.4.2 Message formats**14.2.4.2.1 SecurityParameters message**

The access network sends the SecurityParameters message to provide the access terminal with information that would allow the access terminal to decrypt the content being transmitted on the Forward Broadcast Multicast Services Channel (see Table 396).

Table 396—SecurityParameters message

Field	Length (bits)
MessageID	8
BAKSequenceNumber	4
BlocksPerRandomSeed	8
RandomSeedCount	2
RandomSeedCount + 1 occurrences of the following field:	
RandomSeed	32
Reserved	2

MessageID	The access network shall set this field to 0x00.
BAKSequenceNumber	The access network shall set this field to the sequence number of the Broadcast Access Key that is used to generate the Short Term Key.
BlocksPerRandomSeed	The access network shall set this field to the number of error control blocks encrypted using each Short Term Key.
RandomSeedCount	The access network shall set this field to one less than the number of occurrences of the RandomSeed field following this field in this message.
RandomSeed	The access network shall set this field to the RandomSeed used to generate the Short Term Key.
Reserved	The access terminal shall ignore this field.

The access network shall use the following values for the message information fields when transmitting the SecurityParameters message:

Channels	BCMCSCH (Broadcast Multicast Service Channel)
Addressing	Broadcast

14.2.5 Broadcast Inter-Route Tunneling Protocol

14.2.5.1 Overview

The Broadcast Inter-Route Tunneling Protocol performs the following functions:

- Tunneling over the Forward Broadcast Multicast Services Channel of packets generated by the unicast Route. The Broadcast Inter-Route Tunneling Protocol Header indicates the unicast Route to which the payload belongs.

At the access network, the Broadcast Inter-Route Tunneling Protocol receives packets for transmission from the Route Protocol of a unicast Route. The Broadcast Inter-Route Tunneling Protocol adds a Broadcast Inter-Route Tunneling Protocol Header to the received packet to identify the destination unicast Route and delivers this packet to the Broadcast Packet Consolidation Protocol for transmission.

At the access terminal, the Broadcast Inter-Route Tunneling Protocol receives packets from the Broadcast Packet Consolidation Protocol. The Broadcast Inter-Route Tunneling Protocol removes the Broadcast Inter-Route Tunneling Protocol Header and delivers the packet to the Route Protocol of the corresponding unicast Route.

14.2.5.2 Protocol Data Unit

The protocol data unit for this protocol is a Broadcast Inter-Route Tunneling Protocol Packet. A Broadcast Inter-Route Tunneling Protocol Packet consists of a Broadcast Inter-Route Tunneling Protocol Payload and one or more Broadcast Inter-Route Tunneling Protocol Headers.

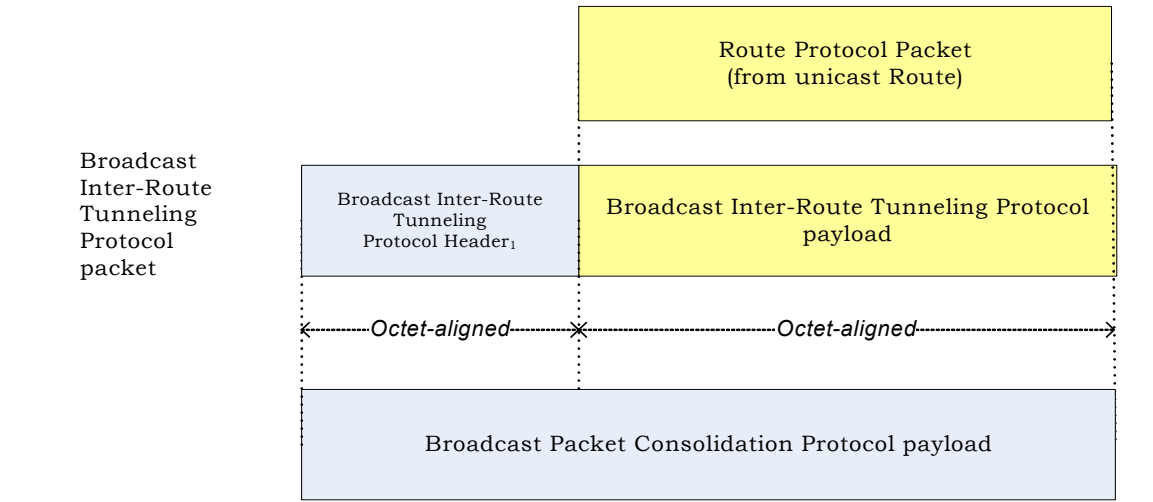


Figure 168—Broadcast Inter-Route Tunneling Protocol Packet

14.2.5.3 Broadcast Inter-Route Tunneling Protocol header

The Broadcast Inter-Route Tunneling Protocol Header has the following format (see Table 397):

Table 397—Broadcast Inter-Route Tunneling Protocol header

Field	Length (bits)
HeaderType	4
If HeaderType= '1000'	
PilotID	10
If HeaderType= '1001'	
ANID	64
Padding	0–7 (as needed)

HeaderType

The access network shall set this field as specified in Table 398 to indicate the type of Broadcast Inter-Route Tunneling Protocol Header.

Table 398—HeaderType values

HeaderType (binary)	Type of Inter-Route Tunneling Protocol header
1000	PilotIDHeader
1001	ANID Header
Other values	Reserved

If the HeaderType field is set to '1000', the access network shall include the following field:

PilotID

The access network shall set this field to the pilot identifier of a pilot belonging to the unicast access network to which this packet belongs.

If the HeaderType field is set to '1001', the access network shall include the following fields:

ANID

The access network shall set this field to the access network identifier (Clause 10) of the unicast access network to which this packet belongs.

Padding

The access network shall include Padding bits to make this Broadcast Inter-Route Tunneling Protocol Header octet-aligned. The access network shall set these bits to 0. The access terminal shall ignore these bits.

14.2.5.4 Access network procedures

If the Broadcast Inter-Route Tunneling Protocol receives a Route Protocol Packet from the Route Protocol of a unicast Route, it shall perform the following:

- The Broadcast Inter-Route Tunneling Protocol shall add a Broadcast Inter-Route Tunneling Protocol Header to identify the unicast access network to which this packet belongs.

- The Broadcast Inter-Route Tunneling Protocol shall deliver the Broadcast Inter-Route Tunneling Protocol Packet to the Broadcast Packet Consolidation Protocol.

14.2.5.5 Access terminal procedures

If the Broadcast Inter-Route Tunneling Protocol receives a Broadcast Inter-Route Tunneling Protocol Packet from the Broadcast Packet Consolidation Protocol, it shall perform the following:

- The Broadcast Inter-Route Tunneling Protocol shall remove the Broadcast Inter-Route Tunneling Protocol Header to produce a Broadcast Inter-Route Tunneling Protocol Payload.
- The Broadcast Inter-Route Tunneling Protocol shall:

deliver the Broadcast Inter-Route Tunneling Protocol payload to the Route Protocol of the unicast Route corresponding to the access network to which this packet belongs.

14.2.6 Broadcast MAC Protocol

14.2.6.1 Overview

The Broadcast MAC Protocol contains the rules governing the operation and the timing of the Forward Broadcast and Multicast Services Channel.

The Broadcast MAC Protocol receives Broadcast Packet Consolidation Protocol packets from the Broadcast Packet Consolidation Protocol. Each packet received from the Broadcast Security Protocol is destined for one Broadcast Logical Channel. The Broadcast MAC Protocol forms an error control block by adding an outer Reed-Solomon code to the payload destined for the logical channel. The Broadcast MAC Protocol thus reduces the radio link error rate as seen by the higher layers.

Figure 169 presents the structure of an Error Control Block. An error control block is formed of N rows and MACPacketSize columns. The top K rows of the error control block contain payload from the served protocols, some of which can be Stuffing Packets. The bottom R = N – K rows of the error control block contain Reed-Solomon parity octets.

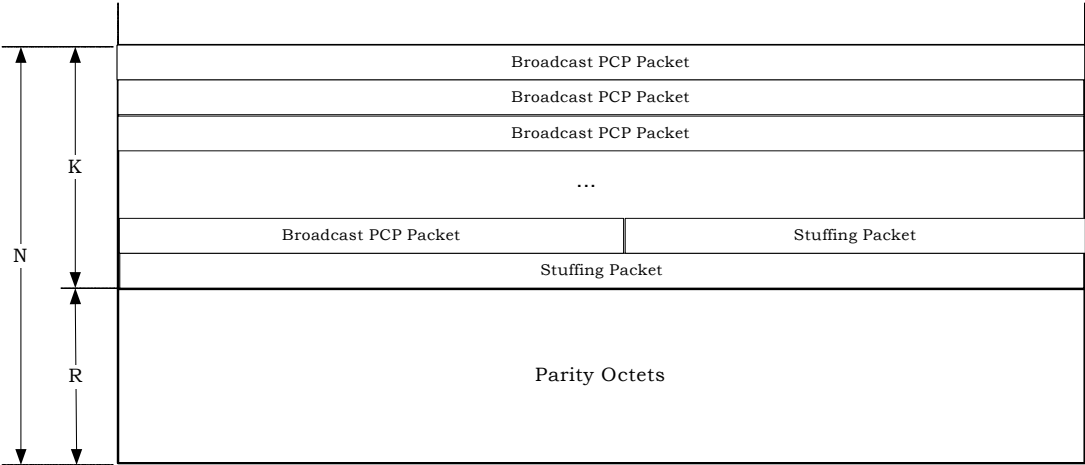


Figure 169—Error Control Block Structure

Data from the error control blocks is transmitted using Broadcast MAC packets. Each row of an error control block is carried using MACPacketSize-bit Broadcast MAC packets. Stuffing packet may be

appended to make the number of data packets equal to K if not enough data is available to fill up the K rows as described in 14.2.6.3.1.6.2.

14.2.6.2 Protocol Data Unit

The protocol data unit for this protocol is a Broadcast MAC packet. Figure 170 illustrates the format of other Broadcast MAC packets except the last non-stuffing packet, which is illustrated in Figure 171. The details of the MAC packet containing the trailer NextUltraframeInfo is described in 14.2.6.3.3.

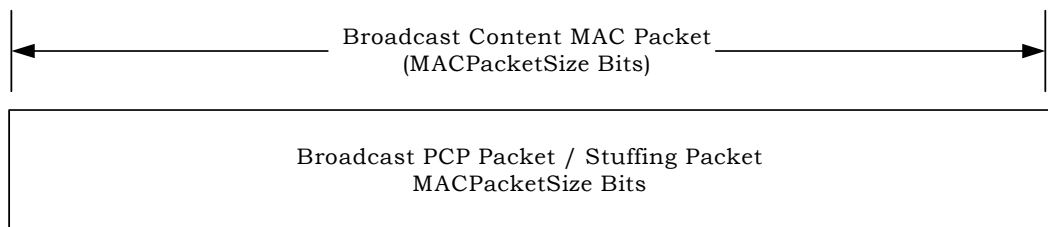


Figure 170—Format of the Broadcast MAC Packets

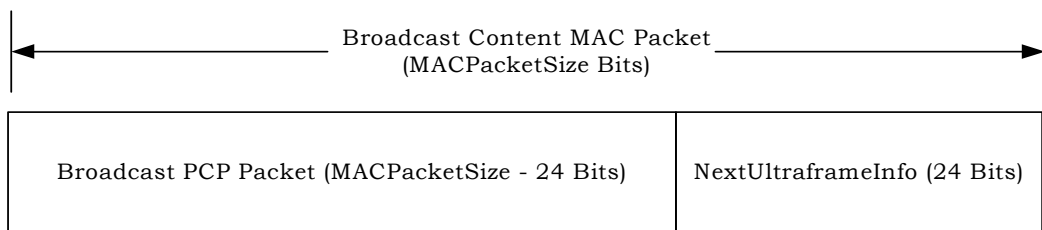


Figure 171—Format of the Last Broadcast MAC Packet that is not a Stuffing Packet

14.2.6.3 Procedures

14.2.6.3.1 Access network requirements

The ultraframe structure of the Forward Broadcast and Multicast Services Channel is described in 14.2.6.3.1.1. The access network shall map the Broadcast Overhead Channel into the Forward Broadcast and Multicast Services Channel as described in 14.2.6.3.1.1.

The access network shall map logical channels into the Forward Broadcast and Multicast Services Channel as described in 14.2.6.3.1.3.

The access network shall transmit the BroadcastParameters message according to the requirements in 14.2.6.3.1.5.

The access network shall create Broadcast MAC packets as follows:

- The access network shall create the error control block as described in 14.2.6.3.1.6.
- The access network shall add the 17-bit NextUltraFrameInfo defined in 14.2.6.3.3 to each Broadcast MAC packet.

The access network shall send the Broadcast MAC packets to the Broadcast Physical Layer Protocol for transmission of the Forward Broadcast and Multicast Services Channel as described in 14.2.6.3.1.6.3.

14.2.6.3.1.1 Broadcast multicast transmission structure

BCMCS transmissions are indexed in terms of ultraframes. Each ultraframe shall consist of a number of subzones and interlaces of 48 Physical Layer super-frames as specified in the 14.2.2.3.4.1.

Information about the physical location of logical channels can be obtained from an associated Broadcast Overhead Channel. Up to four Broadcast Overhead Channels are allowed per ultraframe as specified by NumBOC. The set of Physical channels that each Broadcast Overhead Channel addresses is denoted by a SubbandGroup i , where i can take on values from 0 to 3. The Broadcast Overhead Channels transmitted on ultraframe k shall contain information about the logical channels transmitted on ultraframe $k + 1$.

Each SubbandGroup i shall be partitioned into NumOuterFrames i outerframes, where NumOuterFrames $i = 1, 2, 4, \text{ or } 8$. Each logical channel in an ultraframe shall be transmitted once every outerframe associated with the SubbandGroup i .

14.2.6.3.1.2 BCMCS subband indexing

Over each Physical Layer frame, each group of 128 hop-ports that is part of the Broadcast and Multicast services is referred to as a BCMCS subband in this Standard. The location of these BCMCS subbands is advertised in the BroadcastParameters message. Note that some of these hop-ports may map to guard carriers, and hence not be usable for data transmission.

In each ultraframe, the BCMCS subbands are indexed by UltraframeSubbandIndex are numbered from 0 to NumSubbandsPerUltraframe $- 1$. The PHY frames on which BCMCS is permitted shall be numbered in increasing order with the PHY frame that occurs earlier in time being numbered lower. If more than one BCMCS subband is present in a PHY frame, then each subbands are numbered in increasing order.

As an example, consider a 5 MHz deployment with each BCMCS subband being 128 hop-ports over one PHY frame, represented as a box in Figure 172. The reserved subbands are represented by shaded boxes, while the BCMCS subbands are shaded boxes with an index. This index is referred to as the UltraframeSubbandIndex. In the figure, four subbands are reserved per 8-interlaces, of which three are assigned to BCMCS.

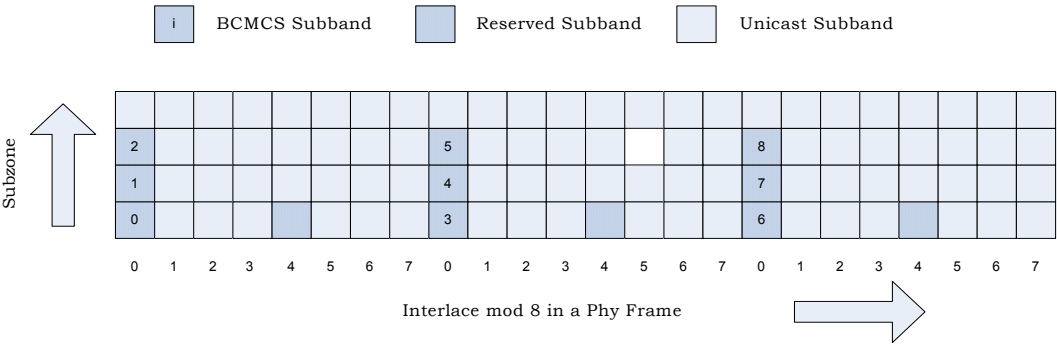


Figure 172—Indexing BCMCS subbands

14.2.6.3.1.3 Indexing BCMCS subbands

Each SubbandGroup i consists of a collection of BCMCS subbands. The i^{th} SubbandGroup is denoted by SubbandGroup i , where i can take on values from 0 to 3. The number of BCMCS subbands in SubbandGroup i is denoted by NumSubbandsPerGroup i , where NumSubbandsPerGroup i is a multiple of NumOuterFrames i .

Define $\text{NumOuterframeSubbandsPerGroup}_i$ to be the integer defined by $\text{NumSubbandsPerGroup}_i / \text{NumOuterFrames}_i$. These BCMCS subbands per Physical Layer group shall be numbered sequentially (in increasing order of the BCMCS subband number) from 0 to $\text{NumOuterframeSubbandsPerGroup}_i - 1$ for each of the SubbandGroup_i corresponding to each outerframe belonging to the SubbandGroup_i . The pair $(\text{OuterframeIndex}_i, \text{SubbandIndex}_i)$ thus has a one-to-one mapping with $\text{UltraframeSubbandIndex}$. Each such BCMCS subband is denoted as $\text{NumSubbandsPerGroup}_i$ is chosen to be a multiple of NumOuterFrames_i .

Each SubbandGroup_i , the BCMCS Subbands assigned to each outerframe are determined by the following pair(s): Offset_j and Period_j . Let k denote the index of BCMCS subband in an outerframe. Every BCMCS subband such that $k = \text{Offset}_j \pmod{\text{Period}_j}$ shall be part of SubbandGroup_i . Note that a SubbandGroup_i can consist of a number of such $(\text{Offset}_j, \text{Period}_j)$ pairs as defined by $\text{NumOffsetsPerGroup}_i$ for $j = 0, 1, \dots, 15$. Note that this can lead to certain unused BCMCS subbands at the end of the superframe, as $\text{NumOuterframeSubbandsPerGroup}_i$ is a multiple of Period_j .

Note that a given $(\text{Offset}_j, \text{Period}_j)$ pair can belong to multiple Physical Channel Groups. In this case, the BroadcastMapping Message of multiple Physical layer Groups can address the same logical channel. This scenario is useful when the SFN coverage of the logical channel is different from the SFN coverage of the Broadcast Overhead Channel.

14.2.6.3.1.4 Logical to Physical Channel Mapping

Each Forward Broadcast and Multicast Services Channel shall consist of a number of BCMCS subbands as specified in the BroadcastParameters message and mapped to logical channels as described in 14.2.6.3.1.1. For all other BCMS subbands, the logical channel occupies all the OFDM symbols.

A logical channel shall carry Broadcast PCP packets from one or more BCMCS Flows. While the same BCMCS Flow may be transmitted independently on several logical channels, the contents of a given BCMCS Flow shall not be split across multiple logical channels. If a BCMCS Flow is carried on more than one logical channel belonging to different sectors, the BCMCS Flow to physical channel mapping need not be the same on all those sectors. Logical channels carrying the same broadcast content may be transmitted synchronously across multiple sectors to facilitate soft combining. A logical channel associated with the Forward Broadcast and Multicast Services Channel may be transmitted synchronously across multiple sectors as described in 14.2.6.3.1.6.5.

14.2.6.3.1.5 Broadcast Overhead Channel

Each sector of an access network can carry up to a maximum of four Broadcast Overhead Channels as defined by the NumBOC parameter. The Broadcast Overhead Channel is sent on the last one, two, four or eight OFDM symbols of each outerframe of a SubbandGroup.

The modulation parameters of the Broadcast Overhead Channel are carried in the BroadcastParameters message. In addition to the Broadcast Overhead Channel, each logical channel also carries in band information about its location in the next ultraframe as described in 14.2.6.3.3.

14.2.6.3.1.6 Generation of Error Control Blocks

14.2.6.3.1.6.1 Overview

The access network shall segment the transmission on a logical channel into error control blocks (ECB). Each error control block shall begin with zero or one MAC packet received by the BCMCS MAC.

The access network shall fill data into the error control block in rows. The access network shall apply Reed-Solomon coding along columns of the error control block. The access network shall transmit the error control block on the Forward Broadcast and Multicast Services Channel in rows.

Each Error Control block shall contain N rows and $MACPacketSize$ columns. The top K rows of the error control block shall contain payload from the served protocols or stuffing packets as described in 14.2.6.3.1.6.2. The bottom $R = N - K$ rows of the error control block shall contain Reed-Solomon parity octets. The length of each Reed-Solomon codeword shall be N octets. Each error control block shall consist of one Reed-Solomon codeword.

The Reed-Solomon code is specified as a (N, K, R) code. N , K , and R are defined as follows:

- a) N = Number of octets in a Reed-Solomon codeword.
- b) K = Number of data octets in a Reed-Solomon codeword.
- c) $R = N - K$ = Number of parity octets in a Reed-Solomon codeword.

Each row of the error control block shall form the payload for one or more Broadcast MAC packets.

A logical channel shall use error control blocks with the same values of N , K , and $MACPacketSize$ on all sectors that the access terminal is allowed to soft combine the logical channel.

14.2.6.3.1.6.1.1 Outer block encoding

The outer code is a Reed-Solomon block code that uses 8-bit symbols and operates in the Galois field called $GF(2^8)$. The primitive element for this field is defined by Equation (31).

$$\alpha^8 + \alpha^4 + \alpha^3 + \alpha^2 + 1 = 0 \quad (31)$$

The j^{th} code symbol ($j = 0, 1, \dots, N - 1$), v_j , shall be defined by Equation (32).

$$v_j = \begin{cases} u_j & 0 \leq j \leq K - 1 \\ \sum_{i=0}^{K-1} u_i * p_{i,j} & K \leq j \leq N - 1 \end{cases} \quad (32)$$

where

N and K are parameters of the (N, K, R) Reed-Solomon code

u_j is the j^{th} of a block of K information symbols

$p_{i,j}$ is the entry on the i^{th} row and the j^{th} column in the parity matrix of the code

$*$ and Σ indicate multiplication and summation in $GF(2^8)$, respectively

14.2.6.3.1.6.1.1.1 (1, 1, 0) Reed-Solomon code

The (1, 1, 0) code generates 1 code symbol for each information symbol input to the encoder. The code symbol shall be the same as the information symbol.

14.2.6.3.1.6.1.1.2 (16, 12, 4) Reed-Solomon code

The (16, 12, 4) code generates 16 code symbols for each block of 12 information symbols input to the encoder. The first 12 symbols are the information symbols and the remaining 4 symbols are parity symbols.

The generator polynomial for the (16, 12, 4) code is

$$g(X) = 1 + \alpha^{201}X + \alpha^{246}X^2 + \alpha^{201}X^3 + X^4$$

The parity matrix for the (16, 12, 4) Reed-Solomon block code shall be as specified in Table 399.

Table 399—Parity Matrix for the (16, 12, 4) Outer Code

Row index <i>i</i>	$P_{i,12}$	$P_{i,13}$	$P_{i,14}$	$P_{i,15}$
0	40	138	141	8
1	8	196	97	158
2	158	4	250	209
3	209	123	27	76
4	76	226	198	160
5	160	142	95	125
6	125	19	59	70
7	70	87	39	137
8	137	169	244	254
9	254	192	27	160
10	160	57	53	201
11	201	246	201	0

NOTE—This table lists the power h of the entry on the i^{th} row and the j^{th} column in the parity matrix, $p_{i,j} = \alpha^h$, where α is the primitive element of GF(256) and $i = 0, \dots, 11$, and $j = 12, 13, 14$, and 15. For example, the entry of 40 in the upper left-hand corner indicates $p_{0,12} = \alpha^{40}$.

14.2.6.3.1.6.1.1.3 (16, 13, 3) Reed-Solomon code

The (16, 13, 3) code generates 16 code symbols for each block of 13 information symbols input to the encoder. The first 13 symbols are the information symbols and the remaining 3 symbols are parity symbols.

The generator polynomial for the (16, 13, 3) code is

$$g(X) = 1 + \alpha^{197}X + \alpha^{197}X^2 + X^3$$

The parity matrix for the (16, 13, 3) Reed-Solomon block code shall be as specified in Table 400.

Table 400—Parity Matrix for the (16, 13, 3) Outer Code

Row index <i>i</i>	$P_{i,13}$	$P_{i,14}$	$P_{i,15}$
0	169	69	236
1	236	34	140
2	140	28	32

Table 400—Parity Matrix for the (16, 13, 3) Outer Code (continued)

Row index i	P _{i,13}	P _{i,14}	P _{i,15}
3	32	88	182
4	182	51	58
5	58	163	238
6	238	175	231
7	231	80	223
8	223	195	250
9	250	237	160
10	160	53	246
11	246	98	197
12	197	197	0

NOTE—This table lists the power h of the entry on the i^{th} row and the j^{th} column in the parity matrix, $p_{i,j} = \alpha^h$, where α is the primitive element of GF(256) and $i = 0, \dots, 12$, and $j = 13, 14$, and 15 . For example, the entry of 169 in the upper left-hand corner indicates $p_{0,13} = \alpha^{169}$.

14.2.6.3.1.6.1.1.4 (16, 14, 2) Reed-Solomon Code

The (16, 14, 2) code generates 16 code symbols for each block of 14 information symbols input to the encoder. The first 14 symbols are the information symbols and the remaining 2 symbols are parity symbols.

The generator polynomial for the (16, 14, 2) code is

$$g(X) = 1 + \alpha^{152}X + X^2$$

The parity matrix for the (16, 14, 2) Reed-Solomon block code shall be as specified in Table 401.

Table 401—Parity Matrix for the (16, 14, 2) Outer Code

Row index i	P _{i,14}	P _{i,15}
0	1	65
1	65	68
2	68	224
3	224	215
4	215	119
5	119	91
6	91	44
7	44	84
8	84	36

Table 401—Parity Matrix for the (16, 14, 2) Outer Code (continued)

Row index i	$P_{i,14}$	$P_{i,15}$
9	36	111
10	111	201
11	201	197
12	197	152
13	152	0
NOTE—This table lists the power h of the entry on the i^{th} row and the j^{th} column in the parity matrix, $p_{i,j} = \alpha^h$, where α is the primitive element of GF(256) and $i = 0, \dots, 13$, and $j = 14$ and 15 . For example, the entry of 1 in the upper left-hand corner indicates $p_{0,14} = \alpha^1$.		

14.2.6.3.1.6.1.1.5 (32, 24, 8) Reed-Solomon code

The (32, 24, 8) code generates 32 code symbols for each block of 24 information symbols input to the encoder. The first 24 symbols are the information symbols and the remaining 8 symbols are parity symbols.

The generator polynomial for the (32, 24, 8) code is

$$g(X) = 1 + \alpha^{44}X + \alpha^{231}X^2 + \alpha^{70}X^3 + \alpha^{235}X^4 + \alpha^{70}X^5 + \alpha^{231}X^6 + \alpha^{44}X^7 + X^8$$

The parity matrix for the (32, 24, 8) Reed-Solomon block code shall be as specified in Table 402.

Table 402—Parity Matrix for the (32, 24, 8) Outer Code

Row index i	$P_{i,24}$	$P_{i,25}$	$P_{i,26}$	$P_{i,27}$	$P_{i,28}$	$P_{i,29}$	$P_{i,30}$	$P_{i,31}$
0	120	75	145	26	65	36	140	205
1	205	104	114	214	181	51	52	211
2	211	207	161	201	132	185	85	141
3	141	128	179	163	34	51	134	89
4	89	78	120	201	16	228	20	158
5	158	165	209	26	193	94	81	183
6	183	33	95	169	72	70	1	43
7	43	83	243	80	240	229	2	243
8	243	22	117	52	230	221	240	68
9	68	168	2	127	148	157	178	252
10	252	145	45	164	120	227	11	87
11	87	174	122	52	2	44	181	20

Table 402—Parity Matrix for the (32, 24, 8) Outer Code (continued)

Row index i	P _{i,24}	P _{i,25}	P _{i,26}	P _{i,27}	P _{i,28}	P _{i,29}	P _{i,30}	P _{i,31}
12	20	5	147	125	141	177	249	186
13	186	6	46	218	27	129	195	67
14	67	55	185	0	3	153	30	151
15	151	207	250	155	56	145	70	2
16	2	39	150	223	214	201	65	45
17	45	169	6	147	51	128	145	64
18	64	100	24	146	118	108	215	32
19	32	191	27	236	189	247	12	174
20	174	93	52	173	213	252	85	160
21	160	240	214	203	155	26	95	238
22	238	22	157	161	236	19	175	44
23	44	231	70	235	70	231	44	0

NOTE—This table lists the power h of the entry on the i^{th} row and the j^{th} column in the parity matrix, $p_{i,j} = \alpha^h$, where α is the primitive element of GF(256) and $i = 0, \dots, 23$, and $j = 24, \dots, 31$. For example, the entry of 120 in the upper left-hand corner indicates $p_{0,24} = \alpha^{120}$.

14.2.6.3.1.6.1.1.6 (32, 26, 6) Reed-Solomon code

The (32, 26, 6) code generates 32 code symbols for each block of 26 information symbols input to the encoder. The first 26 symbols are the information symbols and the remaining 6 symbols are parity symbols.

The generator polynomial for the (32, 26, 6) code is

$$g(X) = 1 + \alpha^{36}X + \alpha^{250}X^2 + \alpha^{254}X^3 + \alpha^{250}X^4 + \alpha^{36}X^5 + X^6$$

The parity matrix for the (32, 26, 6) Reed-Solomon block code shall be as specified in Table 403.

Table 403—Parity Matrix for the (32, 26, 6) Outer Code

Row index i	P _{i,26}	P _{i,27}	P _{i,28}	P _{i,29}	P _{i,30}	P _{i,31}
0	243	66	154	39	233	16
1	16	58	74	113	204	81
2	81	63	43	10	0	29
3	29	210	130	61	234	162
4	162	185	49	175	57	168

Table 403—Parity Matrix for the (32, 26, 6) Outer Code (continued)

Row index i	$P_{i,26}$	$P_{i,27}$	$P_{i,28}$	$P_{i,29}$	$P_{i,30}$	$P_{i,31}$
5	168	182	143	213	35	110
6	110	126	78	245	11	26
7	26	60	14	172	35	249
8	249	126	98	3	112	168
9	168	235	50	228	84	131
10	131	192	197	218	92	141
11	141	13	12	223	195	7
12	7	182	247	197	104	14
13	14	128	241	2	158	3
14	3	15	67	131	98	192
15	192	39	244	247	7	167
16	167	182	222	123	77	30
17	30	113	66	57	164	56
18	56	120	141	45	242	32
19	32	183	185	157	12	147
20	147	210	44	252	175	223
21	223	154	155	195	99	215
22	215	161	30	237	228	70
23	70	177	61	136	39	223
24	223	34	79	169	195	36
25	36	250	254	250	36	0

NOTE—This table lists the power h of the entry on the i^{th} row and the j^{th} column in the parity matrix, $p_{i,j} = \alpha^h$, where α is the primitive element of GF(256) and $i = 0, \dots, 25$, and $j = 26, \dots, 31$. For example, the entry of 243 in the upper left-hand corner indicates $p_{0,26} = \alpha^{243}$.

14.2.6.3.1.6.1.1.7 (32, 28, 4) Reed-Solomon code

The (32, 28, 4) code generates 32 code symbols for each block of 28 information symbols input to the encoder. The first 28 symbols are the information symbols and the remaining 4 symbols are parity symbols.

The generator polynomial for the (32, 28, 4) code is

$$g(X) = 1 + \alpha^{201}X + \alpha^{246}X^2 + \alpha^{201}X^3 + X^4$$

The parity matrix for the (32, 28, 4) Reed-Solomon block code shall be as specified in Table 404.

Table 404—Parity Matrix for the (32, 28, 4) Outer Code

Row index i	$P_{i,28}$	$P_{i,29}$	$P_{i,30}$	$P_{i,31}$
0	207	34	22	229
1	229	210	95	141
2	141	50	89	32
3	32	160	127	224
4	224	37	223	248
5	248	5	131	120
6	120	229	44	228
7	228	73	240	113
8	113	215	118	88
9	88	208	113	74
10	74	37	215	178
11	178	76	97	78
12	78	225	181	5
13	5	218	168	182
14	182	188	204	212
15	212	157	221	40
16	40	138	141	8
17	8	196	97	158
18	158	4	250	209
19	209	123	27	76
20	76	226	198	160
21	160	142	95	125
22	125	19	59	70
23	70	87	39	137
24	137	169	244	254
25	254	192	27	160
26	160	57	53	201
27	201	246	201	0

NOTE—This table lists the power h of the entry on the i^{th} row and the j^{th} column in the parity matrix, $p_{i,j} = \alpha^h$, where α is the primitive element of GF(256) and $i = 0, \dots, 27$, and $j = 28, \dots, 31$. For example, the entry of 207 in the upper left-hand corner indicates $p_{0,28} = \alpha^{207}$.

14.2.6.3.1.6.2 Stuffing packets

The access network shall add stuffing packets to Broadcast PCP packets if necessary to make the payload equal to K rows as described in 14.2.6.3.1.6.1. These packets shall contain an all zero payload and shall not be passed to the Physical layer, and hence not transmitted over the air.

14.2.6.3.1.6.3 Multiplexer

Each row of the error control block shall form the payload for Broadcast MAC packets for a given logical channel, which shall be transmitted in Broadcast Physical Layer packets assigned to the logical channel in time order of the start of transmission of the Broadcast Physical Layer packets.

14.2.6.3.1.6.4 Transmission formats

For each Broadcast Physical Layer packet, the Broadcast MAC Protocol shall provide a *transmission format* to the Broadcast Physical Layer Protocol. A transmission format defines a set of parameters of a Broadcast Physical Layer packet as described in this subclause. The access network shall assign a transmission format to each logical channel. For all Broadcast Physical Layer packets associated with a given logical channel, the access network shall use the transmission format that is associated with that logical channel.

The transmission format of a Broadcast Physical Layer packet shall be of one of the following two types:

- The Broadcast Overhead Channel Transmission
- The Broadcast Transmission Formats

The BroadcastMapping Message transmitted on the Forward Broadcast and Multicast Services Channel is associated with the transmission format of the BroadcastMapping Message (see 14.2.6.3.2.1). A logical channel transmitted on the Forward Broadcast and Multicast Services Channel is associated with the broadcast transmission format (see 9.5).

A basic broadcast transmission format is identified by its packet format index. Each packet format index corresponds to a packet size, rate set, and radio configuration, and modulation order.

The transmission formats for the Broadcast Overhead Channel are shown in Table 405. The spectral efficiencies correspond to the number of outerframes per ultraframe. The modulation order is two in all cases. Note that the number of OFDM symbols (NumOFDMSymbolsPerBOC) required for the transmission of the Broadcast Overhead Channel can be found by $\text{NumOFDMSymbolsPerBOC} = \text{Packet Size} / 99$.

Table 405—Broadcast Overhead Channel transmission formats

Packet format index	Packet size	Rate set	Radio configuration	Spectral efficiency (per outerframes in an ultraframe)			
				1	2	4	8
0	99	1/2/3/4	1	1.021	0.510	0.255	0.128
1	99	1/2/3/4	2	0.421	0.211	0.105	0.053
2	198	1/2/3/4	1	1.021	0.510	0.255	0.128
3	198	1/2/3/4	2	0.421	0.211	0.105	0.053
4	495	1/2/3/4	1	1.021	0.510	0.255	0.128
5	495	1/2/3/4	2	0.421	0.211	0.105	0.053

Table 405—Broadcast Overhead Channel transmission formats (continued)

Packet format index	Packet size	Rate set	Radio configuration	Spectral efficiency (per outerframes in an ultraframe)			
				1	2	4	8
6	792	1/2/3/4	1	1.021	0.510	0.255	0.128
7	792	1/2/3/4	2	0.421	0.211	0.105	0.053

For BCMCS data, the number of bits (including PAD and FCS bits) carried by a Broadcast Physical Layer packet is called its *Packet Size* and denoted by N_{data} . The *Span* of a packet is defined to be the number of transmissions that are allowed for the packet.

A *Rate Set* within a given Transmission Mode is defined by two parameters: The BCMCS subband configuration of the first BCMCS subband of a packet and the packet size. A broadcast transmission format, also called the *BCMCS Transmission Format*, is defined by a Packet Format Index and the number of transmissions. Within each transmission mode, there are four rate sets referred to as Rate Sets 1, 2, 3, 4. Each rate set contains two Radio Configurations in addition to the unicast numerology, which can be used for the third transmission. In this case, the first and second transmissions are sent using broadcast numerology, while the third transmission occurs with unicast numerology.

The Transmission Formats are shown in Table 406. Two BCMCS Transmission Formats are *rate-compatible* if and only if they have the same packet format index, but have possibly different Spans.

The packet format consists of four bits. These four bits index the spectral efficiency and the modulation format to be used for each HARQ transmission of a data packet. Note that the default maximum number of transmissions is three (in this case, the Span is three), however a sector can choose to transmit once or twice as well for each packet format index. In this case, its Span is set to one or two respectively.

The packet size for each assignment is a computed based on the spectral efficiency listed in the table below and the assignment size. The packet size is given in Table 406.

Table 406—Broadcast transmission formats

Packet format index	Packet size	Rate set	Radio configuration	Modulation Order	Spectral efficiency for each transmission		
					1	2	3
0	1536	1	1	4	2.26	1.13	0.72
1	768	1	1	2	1.13	0.57	0.36
2	2048	2	1	4	3.02	1.51	0.96
3	1024	2	1	2	1.51	0.75	0.75
4	2560	3	1	4	1.89	1.26	0.91
5	1280	3	1	2	0.94	0.63	0.46
6	3568	4	1	4	2.64	1.76	1.27
7	1784	4	1	2	1.32	0.88	0.64
8	768	1	2	4	2.18	1.09	0.70

Table 406—Broadcast transmission formats (continued)

Packet format index	Packet size	Rate set	Radio configuration	Modulation Order	Spectral efficiency for each transmission		
					1	2	3
9	384	1	2	2	1.09	0.54	0.35
10	1024	2	2	4	2.90	1.45	0.94
11	512	2	2	2	1.45	0.73	0.47
12	1280	3	2	4	1.82	1.21	0.89
13	640	3	2	2	0.91	0.61	0.44
14	1792	4	2	4	2.54	1.69	1.24
15	896	4	2	2	1.27	0.85	0.62

14.2.6.3.1.6.5 Forward Broadcast and Multicast Services Channel Transmission from multiple sectors

This subclause provides an overview of single frequency network (SFN) and variable-rate transmission on the Forward Broadcast and Multicast Services Channel.

14.2.6.3.1.6.5.1 Single-frequency Network (SFN) Transmission

The Forward Broadcast and Multicast Services Channel is particularly suitable for SFN transmissions in which all sectors in a given broadcast coverage area synchronize their broadcast transmissions and transmit the same waveform (with the exception of sector-dependent delay and complex gain) over the air during the time intervals allocated to the Broadcast Physical Layer packets. At the access terminal's antenna, all transmissions that arrive from the participating sectors combine to appear as a single transmission that goes through a multipath channel with possibly large delay spread between the first and the last arriving paths.

14.2.6.3.1.6.5.2 Rate compatible transmissions

The access network may be configured so that adjacent sectors broadcasting the same content on the same BCMCS subbands may use transmission formats with different spans as long as the transmission formats are rate-compatible with one another. As in the SFN transmission, the transmission time of the first BCMCS subband of each packet are synchronized across all sectors transmitting the same content. The BroadcastParameters message transmitted by a sector specifies the transmission format of the corresponding logical channel. Figure 173 shows a rate compatible transmission scheme, in which Cell A uses a transmission format with a span of 2 BCMCS subbands, while cells B and C use a rate-compatible transmission format with a span of 3 BCMCS subbands. Cell A may choose to transmit a supercast packet in the free slot as illustrated.

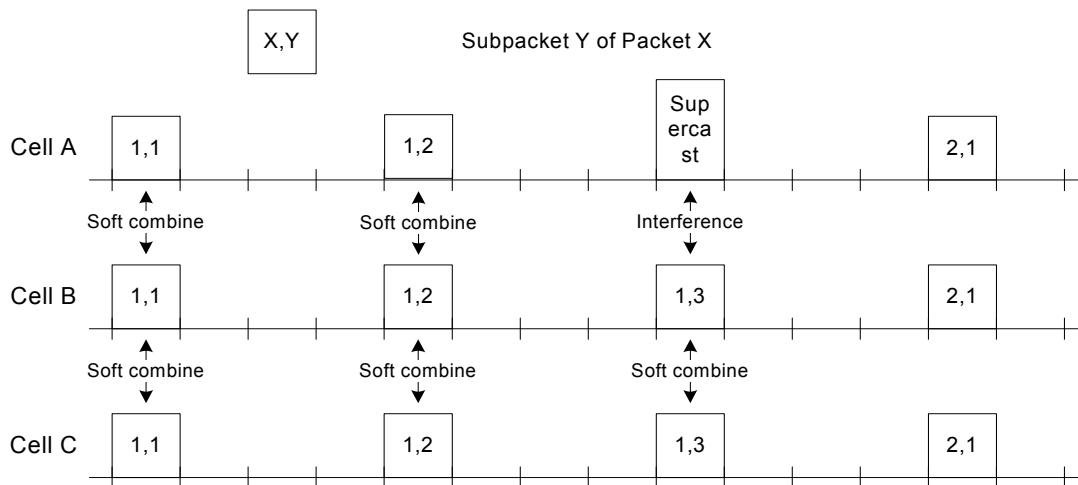


Figure 173—An example of variable rate transmission scheme

14.2.6.3.1.6.6 Forward Broadcast and Multicast Services Channel Parameters

The access network shall determine the values of the following parameters for each logical channel transmitted on the Forward Broadcast and Multicast Services Channel:

- *Period*: The Period parameter assigned to a logical channel shall be greater than or equal to the Span of the transmission format of that logical channel. If the Period is greater than the Span of a Packet Format Index, the access network shall wait until the next period to transmit the next BCMCS packet. The access network may use this available slot(s) for supercast transmissions as illustrated in Figure 173. The method by which the access network sets the Period parameter is beyond the scope of this standard.
- *FDSSeed*: Frequency-domain spreading seed. A 10-bit binary number used in frequency-domain spreading. The method by which the access network sets the FDSSeed parameter is beyond the scope of this standard.
- *PilotStagger*: A parameter used in pilot tone insertion described in 3GPP2 C.S0001. PilotStagger is a content dependent parameter: For different contents broadcast at the same time, the access network should assign different values for the PilotStagger parameter. The method by which the access network sets the PilotStagger parameter is beyond the scope of this standard.
- *Pilot-to-Data Power Ratios*: Specified in 14.2.6.3.1.6.6.1.
- *Modulation Layer*: This parameter specifies whether the logical channel is transmitted on the base layer or the enhancement layer if hierarchical modulation is supported.

For every Broadcast Physical Layer packet, the Broadcast MAC Protocol shall provide the values of the following parameters to the Broadcast Physical Layer Protocol:

- FDSSeed of the logical channel served by the Broadcast Physical Layer packet.
- PilotStagger of the logical channel served by the Broadcast Physical Layer packet.
- DCPDROffset and the in-use pilot-to-data-ratio (PDR) parameters of the logical channel served by the Broadcast Physical Layer packet as defined in 14.2.6.3.1.6.6.1.
- ModulationLayer of the logical channel served by the Broadcast Physical Layer packet.

14.2.6.3.1.6.6.1 Pilot-to-Data Power Ratio (PDR) parameters

The term *in-use pilot-to-data-ratio (PDR) parameters* of a logical channel associated with the Forward Broadcast and Multicast Services Channel shall be represented by 4 bits and interpreted as follows:

Let Z be the value of the 4-bit representation of a parameter when interpreted as an unsigned integer. Then, the value of the parameter in dB is obtained as $0.5 \times (Z - 4)$. For example, a value of -2 dB is represented as '0000' and a value of 5.5 dB is represented as '1111'.

The default Pilot to data Ratios for the Broadcast Overhead Channel associated with the appropriate transmission formats shall be as specified in Table 407.

Table 407—Default Pilot to Data Ratio of the Broadcast Overhead Channel

BOC-Transmission-Format	PDRBOCDefault (dB)
0	2
1	2

For each logical channel associated with the Forward Broadcast and Multicast Services Channel, the access network shall determine the values of the default PDR parameters as follows:

The access network shall determine the values of the default Pilot to Data Ratios of each logical channel based on the BCMCS Transmission Format of that logical channel according to Table 408. The pilot to data ratio for the first two transmissions is denoted by BCMCSPIlotToDataRatio, and may be different from the third transmission, which is denoted by PilotToDataRatio.

The BroadcastParameters message (described in 14.2.6.3.2.1) carries information to indicate the in-use PDR parameters of logical channels.

Table 408—Default Pilot-to-Data Ratio of the Broadcast Messages

BCMCS-Transmission-Format	PDRBOCDefault (dB)
0	2
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	2
9	2
10	2

Table 408—Default Pilot-to-Data Ratio of the Broadcast Messages *(continued)*

BCMCS-Transmission-Format	PDRBOCDefault (dB)
11	2
12	2
13	2
14	2
15	2
16	2

For each logical channel associated with the Broadcast and Multicast Services Channel, the access network shall set the value of a 1-bit parameter called BCMCSPDRProvidedForThisLogicalCh as follows:

- If the BCMCS Transmission Format of the logical channel consists of a span of two or less, the access network shall set the BCMCSPDRProvidedForThisLogicalCh parameter as follows:
 - The access network shall set the BCMCSPDRProvidedForThisLogicalCh parameter to '0' if the value of BCMCSPilotToDataRatio is the same as the value of PDRBOCDefault.
 - Otherwise, the access network shall set the BCMCSPDRProvidedForThisLogicalCh parameter to '1'.
- If the BCMCS Transmission Format of the logical channel consists of a span of 3, the access network shall set the BCMCSPDRProvidedForThisLogicalCh parameter as follows:
 - If all of the following conditions are true, the access network shall set the BCMCSPDRProvidedForThisLogicalCh parameter to '0':
 - The value of BCMCSPilotToDataRatio is the same as the value of PDRBOCDefault for the first two transmissions.
 - The value of PilotToDataRatio is the same as the value of PDRBOCDefault for the third transmissions.
 - Otherwise, the access network shall set the BCMCSPDRProvidedForThisLogicalCh parameter to '1'.

The BroadcastMapping message (described in 14.2.6.3.2.1) carries information to indicate the in-use PDR parameters of logical channels.

14.2.6.3.1.6.7 MAC index

For every Broadcast Physical Layer packet, the Broadcast MAC Protocol shall provide a MAC Index to the Broadcast Physical Layer Protocol. The Broadcast MAC Protocol shall set the MAC Index to

$N_{\text{BroadcastGenericMACIndex}}$

14.2.6.3.2 Message formats**14.2.6.3.2.1 BroadcastMapping message**

The access network sends the BroadcastMapping message to provide the access terminal with the mapping between logical channels and starting BCMCS subband on the Broadcast and Multicast Services Channel (see Table 409).

Table 409—BroadcastMapping message

Field	Length (bits)
MessageID	8
BroadcastParametersSignature	16
NumLogicalChannels	8
NumLogicalChannels occurrences of the following two fields:	
StartLocation j	8
Duration j	4
Padding	0–7 (as needed)

MessageID	The access network shall set this field to 0x00.
NumLogicalChannels	The access network shall set this field to the number of logical channels indexed by the BroadcastMapping message.
StartLocation j	The access network shall set this field to the starting location of logical channel j in number of Subbands i as specified in 14.2.6.3.1.3.
Duration j	The access network shall set this field to the number of number of consecutive BCMCS subbands that the logical channel j occupies as specified in Table 410.
Padding	The access network shall add Padding bits in order to make the length of the entire message equal to an integer number of octets. The access network shall set these bits to zero. The access terminal shall ignore these bits.

**Table 410—Description of the Duration j Field,
where j is the Index of the logical channel**

Duration j	Burst Length (BCMCS subbands)
'0000'	1
'0001'	2
'0010'	3
'0011'	4
'0100'	6
'0101'	8
'0110'	9
'0111'	12

**Table 410—Description of the Duration_j Field,
where *j* is the Index of the logical channel (*continued*)**

Duration _j	Burst Length (BCMCS subbands)
'1000'	16
'1001'	20
'1010'	24
'1011'	32
'1100'	36
'1101'	48
'1110'	60
'1111'	64

14.2.6.3.3 Broadcast MAC NextUltraframeInfo format

The access network shall place the following fields (see Table 411) at the end of every Broadcast MAC packet (see Table 411).

Table 411—Broadcast MAC NextUltraframeInfo format

Field	Length (bits)
StartBCMCSSubbandNumber	8
EndBCMCSSubbandNumber	8
Read BroadcastParameters	1
Reserved	7

StartBCMCSSubbandNumber

The access network shall set this field to the location of the first BCMCS subband of the logical channel in the next ultraframe. If the logical channel is not present in the next ultraframe, the access network shall set this field to NULL.

EndBCMCSSubbandNumber

The access network shall set this field to the location of the last BCMCS subband of the logical channel in the next ultraframe. If the logical channel is not present in the next ultraframe, the access network shall set this field to NULL.

Read BroadcastParameters

The access network shall set this field to '0' if the parameters broadcast on the BroadcastParameters has not changed. The access network shall set this field to '1' if the parameters broadcast on the BroadcastParameters shall change on the next instance of the BroadcastParameters.

Reserved

The access terminal shall ignore this field.

14.2.7 Interface to other protocols**14.2.7.1 Commands Sent**

This protocol does not issue any commands.

14.2.7.2 Indications

This protocol does not register to receive any indications.

14.2.8 Configuration attributes**14.2.8.1 Simple attributes**

This protocol does not define any simple attributes.

14.2.8.2 Complex attributes

This protocol does not define any complex attributes.

14.2.8.3 Non-attribute data

This protocol does not define any static or dynamic non-attribute data.

14.2.8.4 Protocol numeric constants

Protocol numeric constants are listed in Table 412.

Table 412—Protocol numeric constants

Constant	Meaning	Value
$N_{\text{BroadcastType}}$	Type field for this protocol suite	See 16.1
$N_{\text{BroadcastBasicSubtype}}$	Subtype field for this protocol suite	0x0000
$N_{\text{BCIPeriod}}$	The period of repetition of the BroadcastParameters message (in units of superframe)	240

14.2.9 Session State Information

The Session State Information record (see 15.2.6) consists of parameter records. This protocol suite defines the following parameter records:

14.2.9.1 RegisteredBCMCSFlows

The format of the parameter record for the RegisteredBCMCSFlows parameter is shown in Table 413.

Table 413— Format of the parameter record for the RegisteredBCMCSFlows parameter

Field	Length (bits)
ParameterType	8

**Table 413— Format of the parameter record
for the RegisteredBCMCSFlows parameter (continued)**

Field	Length (bits)
Length	8
BCMCSFlowIDLength	2
BCMCSFlowCount	6

BCMCSFlowCount occurrences of the following two fields:

BCMCSFlowID	(BCMCSFlowIDLength + 1) × 8
ExpiryTime	32

ParameterType	This field shall be set to 0x01 for this parameter record.
Length	This field shall be set to the length of this parameter record in units of octets excluding the Length field.
BCMCSFlowIDLength	This field shall be set to one less than the length of the BCMCS Flow Identifier in units of octets. This field shall not be set to '00'.
BCMCSFlowCount	This field shall be set to the number of BCMCS Flow Identifiers included in this parameter record.
BCMCSFlowID	This field shall be set to the BCMCS Flow Identifier that the access terminal may be monitoring.
ExpiryTime	This field shall be set to the System Time in units of 1.28 sec when the registration for the BCMCSFlowID is expected to expire. If the expiry time is unknown, then this field shall be set to zero.

15. Common procedures and data structures

15.1 Common procedures

15.1.1 Protocol initialization for the InConfiguration protocol instance

Upon invocation of the initialization procedures of the InConfiguration instance of a protocol, the access terminal and the access network shall perform the following in the order specified:

- The access terminal and the access network shall set the value of the static data associated with the InConfiguration protocol instance to the corresponding values for the protocol instance of any of the stored personalities.
- If the InConfiguration instance of this protocol is created from a stored Personality, then
 - The access terminal and the access network shall set the values of the dynamic attributes associated with the InConfiguration protocol instance to the values of the corresponding dynamic attributes in the stored Personality.
- If the InConfiguration instance of this protocol is created from a ProtocolSetIdentifier, then
 - The access terminal and the access network shall set value of the dynamic attributes associated with the InConfiguration protocol instance to the default values specified for each dynamic attribute.

15.1.2 Protocol initialization for the InUse protocol instance

Upon invocation of the initialization procedures of the InUse instance of a protocol, the access terminal and access network shall perform the following in the order specified:

- If one or more stored personalities exist, then the access terminal and the access network shall set the value of the static data associated with the InUse protocol instance to the corresponding values in the stored personalities. Otherwise, the access terminal and the access network shall initialize the static attributes associated with the InUse protocol instance to the default values specified for each of them.
- If the InUse instance of this protocol is created from a stored Personality, then
 - The access terminal and the access network shall set value of the dynamic attributes associated with the InUse protocol instance to the values of the corresponding dynamic attributes in the stored Personality.
- If the InUse instance of this protocol is created from an InitialProtocolSetIdentifier, then
 - The access terminal and the access network shall set values of the dynamic attributes associated with the InUse protocol instance to the default values specified for each dynamic attribute.

15.1.3 Hard Commit procedures

The access terminal and the access network shall perform the procedures specified in this subclause, in the order specified, when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures:

- The current InUse protocol instance shall be purged.
- A new InUse protocol instance shall be created from the stored Personality corresponding to the PersonalityIndex field of the SwitchPersonality message.
- Protocol Initialization for the InUse Protocol Instance shall be performed as specified in 15.1.2.

15.1.4 Soft Commit procedures

The access terminal and the access network shall perform the procedures specified in this subclause, in the order specified, when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures:

- The access terminal shall set the static data, static attributes, and soft-committable dynamic attributes of the InUse instance to the corresponding values in the stored Personality corresponding to the InUse Personality.

15.1.5 Hash function

The hash function takes three arguments, Key, N (the number of resources), and Decorrelate (an argument used to de-correlate values obtained for different applications for the same access terminal).

Define:

- Word L to be bits 0–15 of Key
- Word H to be bits 16–31 of Key

where bit 0 is the least significant bit of Key.

The hash value is computed as follows in Equation (33).⁵⁰

$$R = \left\lfloor N \times ((40503 \times (L \oplus H \oplus \text{Decorrelate})) \bmod 2^{16}) / 2^{16} \right\rfloor \quad (33)$$

15.1.6 Pseudorandom number generator

15.1.6.1 General procedures

When an access terminal is required to use the pseudo random number generator described in this subclause, then the access terminal shall implement the linear congruential generator defined by Equation (34).

$$z_n = \alpha \times z_{n-1} \bmod m \quad (34)$$

where

$$\alpha = 7^5 = 16\,807$$

$$m = 2^{31} - 1 = 2\,147\,483\,647$$

z_n is the output of the generator⁵¹

The access terminal shall initialize the random number generator as defined in 15.1.6.2.

The access terminal shall compute a new z_n for each subsequent use.

The access terminal shall use the value $u_n = z_n / m$ for those applications that require a binary fraction u_n , $0 < u_n < 1$.

⁵⁰This formula is adapted from Knuth, D. N., *Sorting and Searching*, vol. 3 of *The Art of Computer Programming*, 3 vols., (Reading, MA: Addison-Wesley, 1973), pp. 508–513. The symbol $\oplus$ represents bitwise exclusive-or function (or modulo 2 addition) and the symbol $\lfloor \rfloor$ represents the “largest integer smaller than” function.

⁵¹This generator has full period, ranging over all integers from 1 to $m - 1$; the values 0 and m are never produced. Several suitable implementations can be found in Park, Stephen K. and Miller, Keith W., “Random Number Generators: Good Ones are Hard to Find,” *Communications of the ACM*, vol. 31, no. 10, October 1988, pp. 1192–1201.

The access terminal shall use the value $k_n = \lfloor N \times z_n / m \rfloor$ for those applications that require a small integer k_n , $0 \leq k_n \leq N - 1$.

15.1.6.2 Initialization

The access terminal shall initialize the random number generator by setting z_0 to

$$z_0 = (\text{HardwareID} \oplus \chi) \bmod m$$

where HardwareID is the least 32 bits of the hardware identifier associated with the access terminal, and χ is a time-varying physical measure available to the access terminal. If the initial value so produced is found to be zero, the access terminal shall repeat the procedure with a different value of χ .

15.1.7 Sequence number validation

When the order in which protocol messages are delivered is important, air interface protocols use a sequence number to verify this order.

The sequence number has s bits. The sequence space is 2^S . All operations and comparisons performed on sequence numbers shall be carried out in unsigned modulo 2^S arithmetic. For any message sequence number N , the sequence numbers in the range $[N + 1, N + 2^{S-1} - 1]$ shall be considered greater than N , and the sequence numbers in the range $[N - 2^{S-1}, N - 1]$ shall be considered smaller than N .

The receiver of the message maintains a receive pointer $V(R)$ whose initialization is defined as part of the protocol. When a message arrives, the receiver compares the sequence number of the message with $V(R)$. If the sequence number is greater than $V(R)$, the message is considered a valid message and $V(R)$ is set to this sequence number; otherwise, the message is considered an invalid message.

15.1.8 AttributeID numbering

AttributeIDs are 16 bits long. AttributeID assignment follows the following rules:

- **0x0000 to 0x00FF**: Simple attributes with one instance. 256 attributes supported.
Example: ConnectionFailureReportingEnabled, which is a Boolean attribute of the Basic Air Link Management Protocol (in the Connection Control Plane).
- **0x01NN to 0x7FNN**: Simple attributes with NN or KK instances. 127 types, with 256 instances per type.
Example ReservationKKStreamFwd, which is an attribute of Basic QoS Management Protocol (in the Radio Link Layer).
- **0x8000 to 0x80FF**: Complex attributes with one instance. 256 attributes.
Example: ANSupportedQoSProfiles, which is an attribute of QoS Management Protocol (in the Radio Link Layer).
- **0x81NN to 0xFFNN**: Complex attributes with NN or KK instances, etc. 127 types, with 256 instances per type.
Example: ReservationKKQoSRequestFwd, which is an attribute of QMP (Radio Link Layer).

15.1.9 Computation of the CRC bits

This subclause describes the computation of CRC bits for a stream of data bits $\{b_{N-1}, b_{N-2}, \dots, b_1, b_0\}$. Here b_0 is the initial bit in the stream and b_{N-1} is the final bit.

The CRC shall be calculated using the standard CRC-CCITT generator polynomial [see Equation (35)].

$$g(x) = x^{24} + x^{23} + x^6 + x^5 + x + 1 \quad (35)$$

The CRC shall be equal to the value computed according to the following procedure as shown in Figure 174

- All shift-register elements shall be initialized to '0's.
- The register shall be clocked once for each data bit. The bit stream shall be read starting from b_0 and ending in b_{N-1} .
- When all the data bits are exhausted, the values remaining in the shift registers $\{c_0, c_1, \dots, c_{23}\}$ constitute the 24-bit CRC. The data stream padded with a 24-bit CRC shall be $\{c_0, c_1, \dots, c_{22}, c_{23}, b_{N-1}, b_{N-2}, \dots, b_1, b_0\}$ i.e., b_0 is the initial bit and c_0 is the last bit in the padded stream.
- If less than a 24-bit CRC is to be used, the most significant bits of the 24-bit CRC shall be used. The data stream padded with a M-bit CRC ($M < 24$) CRC shall be $\{c_{24-M}, c_{24-M+1}, \dots, c_{23}, b_{N-1}, b_{N-2}, \dots, b_1, b_0\}$ i.e., b_0 is the initial bit and c_{24-M} is the last bit in the padded stream.

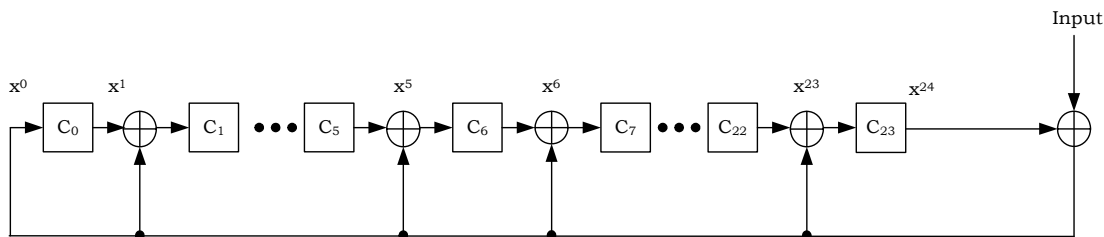


Figure 174—CRC calculation

15.2 Common data structures

15.2.1 ChannelBand record

The ChannelBand record defines an access network channel frequency and the type of system on that frequency. This record contains the fields shown in Table 414.

Table 414—ChannelBand record fields

Field	Length (bits)
SystemType	8
Length	8
SystemTypeSpecificField	$8 \times \text{Length}$

SystemType The access network shall set this field to one of the values shown in Table 415.

Table 415—SystemType encoding

Field value	Meaning
0x00	System compliant to this standard. ChannelNumber field specifies forward channel and reverse channel that are FDD-paired.
0x01	System compliant to 3GPP2 C.S0001. ^a
0x02	System compliant to 3GPP2 C.S0024. ChannelNumber field specifies the forward channel and reverse channel that are FDD-paired.

Table 415—SystemType encoding (continued)

Field value	Meaning
0x03	System compliant to 3GPP2 C.S0024. ChannelNumber field specifies only the forward channel
0x04	Reserved
0x05-0xff	Reserved

^aSystemType of 0x01 applies to 3GPP2 C.S0001 and all of its predecessors.

Length This field shall be set to the length, in units of octets, of SystemTypeSpecific fields.

SystemTypeSpecificFields This field shall be set according to the value of SystemType as specified in 15.2.1.1.

15.2.1.1 SystemTypeSpecificFields when SystemType is 0x00

Table 416—SystemTypeSpecificFields (0x00)

Field	Length (bits)
BandClass	8
ChannelNumber	16
HalfDuplexIncluded	1
HalfDuplexSupported	0 or 1
ReverseChannelIncluded	0 or 1
ReverseChannelBandClass	0 or 8
ReverseChannelNumber	0 or 16
CyclicPrefixIncluded	1
CyclicPrefixLength	0 or 2
FFTSizeIncluded	1
FFTSize	0 or 3
NumGuardSubcarriersIncluded	1
NumGuardSubcarriers	0 or 6
Padding	Variable

BandClass This field shall be set to the band class number corresponding to the frequency assignment of the forward channel specified by this record.

ChannelNumber This field shall be set to the channel number corresponding to the frequency assignment of the forward channel specified by this record.

HalfDuplexIncluded	This field shall be set to '1' if the HalfDuplexSupported and ReverseChannelIncluded fields are included in this record.
HalfDuplexSupported	This field shall be omitted if HalfDuplexIncluded is equal to '0'. Otherwise, this field shall be set to '1' if half-duplex operation is supported on this channel, and to '0' if half-duplex operation is not supported on this channel.
ReverseChannelIncluded	This field shall be omitted if HalfDuplexIncluded is set to '0'. Otherwise, this field shall be included and set as follows: This field shall be set to '1' if ReverseChannelBandClass and ReverseChannelNumber fields are included. Otherwise, this field shall be set to '0'. If this field is set to '0', the reverse channel is as specified by the frequency division duplex pair reverse channel corresponding to the forward channel.
ReverseChannelBandClass	This field shall be omitted if ReverseChannelIncluded is omitted or is set to '0'. Otherwise, this field shall be set to the BandClass of the reverse channel.
ReverseChannelNumber	This field shall be omitted if ReverseChannelIncluded is omitted or is set to '0'. Otherwise, this field shall be set to the BandClass of the reverse channel.
CyclicPrefixIncluded	This field shall determine if the length of the cyclic prefix is included in this record.
CyclicPrefixLength	This field shall be omitted if CyclicPrefixIncluded is equal to '0'. Otherwise, this field shall be set to the CyclicPrefix of the forward channel. The interpretation of this field shall be as defined by the Overhead Messages Protocol.
FFTSizeIncluded	This field shall determine if the FFT size is included in this record.
FFTSize	This field shall be omitted if FFTSizeIncluded is equal to '0'. Otherwise, this field shall be set to $\log_2(N_{\text{FFT}}/128)$.
NumGuardSubcarriersIncluded	This field shall determine if the number of guard carriers is included in this record.
NumGuardSubcarriers	This field shall be omitted if NumGuardCarriersIncluded is equal to '0'. Otherwise, this field shall be set to the NumGuardCarriers of the forward channel. The interpretation of this field shall be as defined by the Overhead Messages Protocol.
Padding	The length of this field shall be set to octet align this record. This field shall be set to zero. The receiver shall ignore this field.

15.2.1.2 SystemTypeSpecificFields when SystemType is 0x01, 0x02, or 0x03**Table 417—SystemTypeSpecificFields (0x01, 0x02, or 0x03)**

Field	Length (bits)
BandClass	5
ChannelNumber	11

BandClass

If the SystemType field is set to 0x01 or 0x02, the access network shall set this field to the band class number corresponding to the frequency assignment of the channel specified by this record for both the forward channel and the reverse channel. If the SystemType is set to 0x03, then access network shall set this field to the band class number corresponding to the frequency assignment of the channel specified by this record for the forward channel only.

ChannelNumber

If the SystemType is set to 0x01 or 0x02, the access network shall set this field to the channel number corresponding to the frequency assignment of the channel specified by this record for both the forward channel and the reverse channel. If the SystemType is set to 0x03, this access network shall set this field to the channel number corresponding to the frequency assignment of the channel specified by this record for the forward channel only.

15.2.2 Neighbor technology record

The Neighbor Technology Record defines an parameters for a neighboring access network belonging to another technology.

If TechnologyType is 0x00 (CDMA2000 1x), then this record contains the fields shown in Table 418.

Table 418—Neighbor technology record (0x0)

Field	Length
ChannelRecord	24
SIDNIDIncluded	1
SID	0 or 15
NID	0 or 16
PacketZoneIDIncluded	1
PacketZoneID	0 or 8
Padding	Variable

ChannelRecord

This field shall be set to the ChannelRecord for the CDMA2000 1x system.

SIDNIDIncluded

This field shall determine if the SID and NID fields are included.

SID

The access network shall omit this field if SIDNIDIncluded is '0'. Otherwise, the field shall be set to the System Identifier for the CDMA2000 1x system.

NID	The access network shall omit this field if SIDNIDIncluded is '0'. Otherwise, the field shall be set to the Network Identifier for the CDMA2000 1x system. The access network may set this field to all ones to indicate an unspecified NID value.
PacketZoneIDIncluded	This field shall determine if a PacketZoneID is included.
PacketZoneID	The access network shall omit this field if PacketZoneIDIncluded is '0'. Otherwise, this field shall be set to the packet zone ID for the CDMA2000 1x system.
Padding	The length of this field shall be set to octet align this record. This field shall be set to zero. The access terminal shall ignore this field.

If TechnologyType is 0x01 (CDMA2000 High Rate Packet Data), then this record contains the fields shown in Table 419.

Table 419—Neighbor technology record (0x01)

Field	Length
ChannelRecord	24
SubnetLength	8
Subnet	SubnetLength
Padding	Variable

ChannelRecord	This field shall be set to the ChannelRecord for the HRPD system.
SubnetLength	This field shall be set to the length of the Subnet of the HRPD system.
Subnet	This field shall be omitted if SubnetLength is zero. Otherwise, this field shall be set to the Subnet for the HRPD system.
Padding	The length of this field shall be set to octet align this record. This field shall be set to zero. The access terminal shall ignore this field.

15.2.3 Access terminal identifier record

The Access Terminal Identifier record provides a unicast, multicast, or broadcast access terminal address. This record contains the fields shown in Table 420.

Table 420—Access terminal identifier record

Field	Length (bits)
ATIType	2
ATI	0 or 128

ATIType Access Terminal Identifier Type. This field shall be set to the type of the ATI, as shown in Table 421:

Table 421—ATIType field encoding

ATIType	ATIType Description	ATI Length (bits)
'00	'Invalid	n/a
'01	'Multicast ATI	128
'10	'Unicast ATI	128
'11	'Random ATI (RATI)	128

ATI Access Terminal Identifier. This field shall be set as shown in Table 421.

15.2.4 Attribute record

The attribute record defines a set of suggested values for a given attribute. The attribute record indicates the Protocol Type to which the attribute belongs. The attribute record format is defined, such that if the recipient does not recognize the attribute, it can discard it and parse attribute records that follow this record.

An attribute can be one of the following three types:

- Simple attribute, if it contains a single value,
- Attribute list, if it contains multiple single values that are to be interpreted as different suggested values for the same attribute identifier (e.g., a list of possible protocol Subtypes for the same protocol Type), or
- Complex attribute, if it contains multiple values that together form a complex value for a particular attribute identifier.

Simple attributes are a special case of an attribute list containing a single value.

The type of the attribute is determined by the attribute identifier.

The sender of a ConfigurationResponse message or FastConfigurationAccept message (see Session Control Protocol) selects an attribute-value from a ConfigurationRequest message or FastConfigurationRequest message respectively by sending the attribute value if it is a simple attribute or a selected value out of an attribute list. Selection of complex-attributes is done by sending the value identifier that identifies the complex value.

The format of a simple attribute and attribute list is given by Table 422.

Table 422—Attribute record (Simple attribute)

Field	Length (bits)
AttributeID	16
One or more instances of the following record	
LengthLength	1
Length	7 or 15

Table 422—Attribute record (Simple attribute) (continued)

Field	Length (bits)
AttributeValue	8 × Length

AttributeID	The sender shall set this field to the attribute identifier of this attribute.
LengthLength	This field shall be set to '0' if the length of the Length field is seven bits. Otherwise, this field shall be set to '1'.
Length	This field shall be set to the length of the AttributeValue field in units of octets.
AttributeValue	A suggested value for the attribute. Attribute value lengths are, in general, an integer number of octets.

The format of a complex attribute is given by Table 423.

Table 423—Attribute record (complex attribute)

Field	Length (bits)
AttributeID	16

One or more instances of the following fields

LengthLength	1
Length	7 or 15
ValueID	Protocol Specific

An appropriate number of instances of the following record for each instance of the ValueID field

AttributeValue	Attribute dependent

AttributeID	The sender shall set this field to the attribute identifier of this attribute.
LengthLength	This field shall be set to '0' if the length of the Length field is seven bits. Otherwise, this field shall be set to '1'.
Length	This field shall be set to the length of the ValueID field plus the length of the AttributeValue field in units of octets.
ValueID	It identifies the set of attribute values following this field. The sender shall increment this field for each new set of values for this complex attribute.
AttributeValue	A suggested value for the attribute. Attribute value lengths are in general an integer number of octets.

15.2.5 Non-attribute data record

Non-attribute data could be included in a Session State Information Record. Format of the non-attribute data is as follows in Table 424.

Table 424—Non-attribute data record

Field	Length (bits)
DataID	16
DataValue	Data dependent
Padding	Variable

DataID	The sender shall set this field to the non-attribute data identifier of this data.
DataValue	A value for the data. Data value lengths are in general an integer number of octets.
Padding	The length of this field is the smallest value that will make the Data record octet aligned. The sender shall set this field to zero. The receiver shall ignore this field.

15.2.6 Session State Information Record

The Session State Information is to be used for transferring the session parameters from a source access network to a target access network. Session parameters are the attributes of the Personalities and the internal parameters that define the state of each InUse instance. The format of this record is shown in Table 425. If an attribute is not contained in the Session State Information Record, the target access network shall assume that the missing attributes have the default values (specified for each attribute in each protocol). The sender shall include all the Parameter Records associated with a (PersonalityID, ProtocolType) pair in the same Session State Information Record.

Table 425—Session State Information Record format

Field	Length (bits)
FormatID	8
PersonalityID	4
Reserved	4
ProtocolSetIdentifier	16
ProtocolType	8 or 16
ProtocolSubtype	8

One or more instances of the following Parameter Record:

ParameterType	8
Length	16
ParameterType-specific record	$8 \times \text{Length}$

FormatID	This field identifies the format of the rest of the fields in this record and shall be set to zero.

PersonalityID	This field shall be set to the identifier associated with the Personality whose Session State Information is being conveyed.
Reserved	This field shall be ignored by the receiver.
ProtocolSetIdentifier	This field shall be set to the Protocol Set Identifier associated with this Personality.
ProtocolType	If the length of this field is 8 bits, then the most significant bit of this field shall be set to '0'; otherwise, the most significant bit of this field shall be set to '1'. This field shall be set to the protocol to which this attribute belongs.
ProtocolSubtype	This field shall be set to the protocol subtype value (see Table 427 and Table 428) for the protocol associated with the encapsulated session parameters.
ParameterType	This field shall be set according to Table 426.

Table 426—ParameterType Field encoding

Field Value	Meaning
0x00	The ParameterType-specific record consists of a Complex or a Simple Attribute as defined in 15.2.4. The ValueID field of the complex attribute shall be set to zero.
0x01	The ParameterType-specific record consists of Non-Attribute Data as defined in by each protocol.
All other values	ParameterType-specific record are protocol dependent

Length Length in octets of the ParameterType-specific-record following this field.

ParameterType-specific record

If the ParameterType field is set to 0x00, then this record shall be set to the simple or complex attribute (see 15.2.4) associated with the protocol identified by the (ProtocolType, ProtocolSubtype) pair. If the ParameterType field is set to 0x01, then this record shall be set to the Non-Attribute Data associated with the protocol identified by the (ProtocolType, ProtocolSubtype) pair. Otherwise, the structure of this record shall be as specified by the protocol, which is identified by the (ProtocolType, ProtocolSubtype) pair.

16. Assigned names and numbers

16.1 Protocol types and subtypes

The Protocol Type may be 8 or 16 bits long. Structurally a Protocol Type consists of two sub-fields; a Type length field (of size 1-bit) and a Type subfield (of size 7 or 15 bits). If a Protocol Type is 8 bits long, its MSB shall be set to '0'. Otherwise, its MSB shall be set to '1'.

The following rules are used for Protocol Type assignment:

- '0'|'ppppppp' is used as Protocol Type for InUse instance of non-Application Layer protocols, where 'ppppppp' are seven binary digits.
- Protocol Types described by '100'|'pppppppppppppp' where 'pppppppppppppp' are thirteen binary digits is reserved.
- '110'|'000000'|'ppppppp' bits is used as Protocol Type for InConfiguration instance of non-Application Layer protocols, and 'ppppppp' are seven binary digits. The same value of 'ppppppp' is used for the InUse and InConfiguration instance of any non-Application Layer protocol.
- '101'|'ttttt'|'aaaaaaaa' is used as Protocol Type for InUse instance of Application Layer protocols, where 'ttttt' is are five binary digits, and 'aaaaaaaa' are eight binary digits representing the Application Layer Protocol ID as specified in Table 427.
- '111'|'ttttt'|'aaaaaaaa' is used as Protocol Type for InConfiguration instance of Application Layer protocols, where 'ttttt' is are five binary digits, and 'aaaaaaaa' are eight binary digits representing the Application Layer Protocol ID as specified in Table 428. The same value of 'ttttt' is used for the InUse and InConfiguration instance of a given Application Layer protocol instantiation.

Table 427—Protocol type and subtype for InUse instance

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Physical Layer	0x00	8	Basic Physical Layer	0x0000
Packet Consolidation	0x01	8	Basic Packet Consolidation	0x0000
Superframe Preamble MAC	0x02	8	Basic Superframe Preamble MAC	0x0000
Access Channel MAC	0x03	8	Basic Access Channel MAC	0x0000
Forward Link Control Segment MAC	0x04	8	Basic Forward Link Control Segment MAC	0x0000
Forward Traffic Channel MAC	0x05	8	Basic Forward Traffic Channel MAC	0x0000
Reverse Traffic Channel MAC	0x06	8	Basic Reverse Traffic Channel MAC	0x0000
Reverse Control Channel MAC	0x07	8	Basic Reverse Control Channel MAC	0x0000
Route	0x09	8	Basic Route	0x0000
Stream	0x0a	8	Basic Stream	0x0000
Radio Link corresponding to Stream NN^a	0x40 + NN	8	Basic Radio Link	0x0000
QoS Management	0x0b	8	Basic QoS Management	0x0000

Table 427—Protocol type and subtype for InUse instance (continued)

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Application	0xa0PP to 0xbfPP ^b	16	NA	NA
Key Exchange	0x0c	8	Basic Key Exchange	0x0000
Ciphering	0x0d	8	AES Ciphering	0x0000
Message Integrity Protection	0x0e	8	Basic Message Integrity	0x0000
Air Link Management	0x0f	8	Basic Air Link Management	0x0000
Initialization State	0x10	8	Basic Initialization State	0x0000
Idle State	0x11	8	Basic Idle State	0x0000
Connected State	0x12	8	Basic Connected State	0x0000
Active Set Management	0x13	8	Basic Active Set Management	0x0000
Overhead Message	0x14	8	Basic Overhead Message	0x0000
Session Control	0x15	8	Basic Session Configuration	0x0000
Route Control	0x17	8	Basic Route Control	0x0000
BCMCS Protocol Suite	0x18	8	Basic BCMCS Suite	0x0000

^aNN is the two-digit hexadecimal Stream number in the range 0x00 to 0x1f.

^bPP is the two-digit hexadecimal Application Layer ProtocolID in the range 0x00 to 0xff.

Table 428—Protocol type and subtype for InConfiguration instance

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Physical Layer	0xc000	16	Basic Physical Layer	0x0000
Packet Consolidation	0xc001	16	Basic Packet Consolidation	0x0000
Superframe Preamble MAC	0xc002	16	Basic Superframe Preamble MAC	0x0000
Access Channel MAC	0xc003	16	Basic Access Channel MAC	0x0000
Forward Link Control Segment MAC	0xc004	16	Basic Forward Link Control Segment MAC	0x0000
Forward Traffic Channel MAC	0xc005	16	Basic Forward Traffic Channel MAC	0x0000
Reverse Traffic Channel MAC	0xc006	16	Basic Reverse Traffic Channel MAC	0x0000
Reverse Control Channel MAC	0xc007	16	Basic Reverse Control Channel MAC	0x0000

Table 428—Protocol type and subtype for InConfiguration instance (continued)

Protocol Type			Protocol Subtype	
Name	ID	Length (bits)	Name	ID
Route	0xc009	16	Basic Route	0x0000
Stream	0xc00a	16	Basic Stream	0x0000
Radio Link for Stream <i>NN</i>	0xc040 + <i>NN</i>	16	Basic Radio Link	0x0000
QoS Management	0xc00b	16	Basic QoS Management	0x0000
Application	0xe0PP to 0xffPP	16	NA	NA
Key Exchange	0xc00c	16	Basic Key Exchange	0x0000
Ciphering	0xc00d	16	AES Ciphering	0x0000
Message Integrity Protection	0xc00e	16	Basic Message Integrity Protection	0x0000
Air Link Management	0xc00f	16	Basic Air Link Management	0x0000
Initialization State	0xc010	16	Basic Initialization State	0x0000
Idle State	0xc011	16	Basic Idle State	0x0000
Connected State	0xc012	16	Basic Connected State	0x0000
Active Set Management	0xc013	16	Basic Active Set Management	0x0000
Overhead Messages	0xc014	16	Basic Overhead Messages	0x0000
Session Control	0xc015	16	Basic Session Control	0x0000
Route Control	0xc017	16	Basic Route Control	0x0000
BCMCS Protocol Suite	0xc018	16	Basic BCMCS Protocol Suite	0x0000

Table 429—Services Sublayer Protocol ID

Value	Services Sublayer Protocol
0x00	NULL
0x01	Reserved
0x02	Reserved
0x03	Reserved
0x04	Robust Header Compression (ROHC) Support Protocol
0x05	Internet Protocol (IP) version 4 as defined in IETF RFC 791 and version 6 as defined in IETF RFC 2460
0x06	Basic Signaling Protocol
0x07	Basic Inter-Route Tunneling Protocol

Table 429—Services Sublayer Protocol ID *(continued)*

Value	Services Sublayer Protocol
0x08	EAP Support Protocol

Table 430—Initial Protocol Set Identifier

Value	Protocol Stack
0x0	Protocol stack with all other protocols with subtype 0x0000.
All other values	Reserved.

Table 431—Initial Protocol Set Identifier

BCMCS IPSI Value	Protocol Stack
0x0	Protocol stack consisting of BCMCS Protocol Suite with subtype 0x0000.
All other values	Reserved.

Table 432—Protocol Set Identifier

Value	Protocol Stack
0x0000	Protocol stack with all other protocols with subtype 0x0000.
0xZff0–0xZfff, where Z is any value of IPSI	Reserved for vendor-specific assignment.
All other values	Reserved.

16.2 ANID, SectorID, and UATI provisioning

The access network shall follow the procedures in this subclause to ensure that the ANID, SectorID, and UATI are globally unique. Unless specified otherwise, the access terminal shall not assume any structure in the ANID, SectorID, and UATI.

16.2.1 ANID Construction

Access network shall construct a globally Unique ANIDs as follows:

Globally Unique ANID shall be set to a 6to4 IPv6 address as specified in Figure 175 (see IETF RFC 3056).

FP '001' (3 bits)	TLA '00000000000010' (13 bits)	V4ADDR (32 bits)	SLA ID (16 bits)	0x00000000000000001 (64 bits)

Figure 175—ANID Construction as a 6to4 IPv6 Address

where

- V4ADDR is the globally routable IPv4 address of a group of access networks (group size may be one).
- SLA ID is a locally unique ID given to this access network within the group of access networks sharing V4ADDR.

16.2.2 SectorID construction

Access network shall construct a globally unique SectorID as illustrated in Figure 176:

64 MSBs of ANID to which this Sector belongs (64 bits)	ID of the Sector unique within the AN (64 bits)

Figure 176—Sector ID Construction

64 LSBs of SectorID shall not be set to any of the following:

- 0x0000000000000000, or
- 0x0000000000000001, or
- 64 LSBs of another SectorID with the same 64 MSBs as this SectorID, or
- 64 LSBs of another UATI with the same 64 MSBs as this SectorID.

16.2.3 UATI construction

Access network shall construct a globally unique UATI as illustrated in Figure 177.

64 MSBs of ANID to which this UATI belongs (64 bits)	ID of the AT unique within the AN (64 bits)

Figure 177—UATI construction

64 LSBs of UATI shall not be set to any of the following:

- 0x0000000000000000, or
- 0x0000000000000001, or
- 64 LSBs of another SectorID with the same 64 MSBs as this UATI, or
- 64 LSBs of another UATI with the same 64 MSBs as this UATI.

17. MAC and PHY MIB

17.1 Overview

This clause defines a Management Information Base (MIB) module for managing the MAC and PHY. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). The objects in this MIB are defined using the mechanisms specified in the Structure of Management Information (SMI). The MIB module specified is compliant to SMIV2, which is described in IETF RFC 2578, IETF RFC 2579, and IETF RFC 2580.

17.2 MIB structure

The MIB structure is based on the architecture reference model in Figure 1 and the layering architecture for the air interface in Figure 2. The MIB object is composed of two groups:

- dot20An: This group contains managed objects defined for the access network.
- dot20Cmn: This group contains managed objects defined for the access network and the access terminal.

17.3 Definition

```
IEEE802dot20-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    ifIndex
        FROM IF-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    Counter32, Counter64, Integer32, MODULE-IDENTITY, OBJECT-IDENTITY,
    OBJECT-TYPE, transmission
        FROM SNMPv2-SMI
    RowPointer, RowStatus, TEXTUAL-CONVENTION, TruthValue
        FROM SNMPv2-TC
    ;
```

```
ieee802dot20 MODULE-IDENTITY
```

```
    LAST-UPDATED "200703301948Z" -- March 30, 2007
```

```
    ORGANIZATION
```

```
        "IEEE 802.20"
```

```
    CONTACT-INFO
```

```
        "WG URL: ieee802.org/20"
```

```
        MIB Contact: J. Tomcik
```

```
        Postal: 5775 Morehouse Dr
```

```
        San Diego, CA, 92121, USA
```

```
        Tel: 858-658-3231
```

```
        Fax: 858-658-3231
```

```
        E-mail: jtomcik@qualcomm.com"
```

```
    DESCRIPTION
```

```
        "The MIB module for IEEE 802.20 entities.
```

```
    )"
```

```
 ::= { iso (1) iso-identified-organization (3) ieee (111) standards-
association-numbered-series-standards (2) lan-man-stds (802) ieee802dot20
(20) ieee802dot20mibs(1) ieee802dot20widebandMIB(1) }
```

```
Dot20AnChannelBandsEntry ::= SEQUENCE
```

```
{
    dot20AnChannelBandIndex      Integer32,
    dot20AnChannelBandClass      Integer32,
    dot20AnChannelNumber         Integer32,
    dot20AnReverseChannelBandClass Integer32,
    dot20AnReverseChannelNumber  Integer32,
    dot20AnChannelBandShortId    Integer32,
    dot20AnChannelBandAccessHashMask Integer32,
    dot20AnChannelBandStatus     RowStatus
}
```

```
Dot20AnIdleStateStatsEntry ::= SEQUENCE
```

```
{
    dot20AnAccessAttemptCounts      Counter32,
    dot20AnAccessAttemptFailCounts Counter32,
    dot20AnPageAttemptCounts        Counter32,
    dot20AnPageFailureCounts         Counter32
}
```

```
Dot20AnNeighborListEntry ::= SEQUENCE
```

```
{
    dot20AnNeighborIndex      Integer32,
    dot20AnNeighborSectorPointer RowPointer,
    dot20AnNeighborRowStatus   RowStatus
}
```

```
Dot20AnNeighborSectorsEntry ::= SEQUENCE
```

```
{
    dot20AnNeighborSectorIndex      Integer32,
    dot20AnNeighborSectorID          OCTET STRING,
    dot20AnNeighborPilotPN          Integer32,
    dot20AnNeighborTransmitPower     Integer32,
    dot20AnNeighborGloballySynch     TruthValue,
    dot20AnNeighborSectorAnGroupId   Integer32,
    dot20AnNeighborSectorPilotGrpId  Integer32,
    dot20AnNeighborSynchGroupId      Integer32,
    dot20AnNeighborSectorCellGroupId Integer32,
    dot20AnNeighborSectorStatus      RowStatus
}
```

```
Dot20AnOtherTechNghbrsEntry ::= SEQUENCE
```

```
{
    dot20AnOtherTechnologyIndex      Integer32,
    dot20AnTechnologyType            Integer32,
    dot20AnTechNghbrListLength       Integer32,
    dot20AnTechnologyNeighborList    OCTET STRING,
    dot20AnOtherTechNghbrRowStatus   RowStatus
}
```

```
Dot20AnSecondaryRegZoneCodeEntry ::= SEQUENCE
```

```
{
    dot20AnSecondaryRegZoneCodeIndex Integer32,
    dot20AnSecRegZoneCode             Integer32,
    dot20AnSecondaryRegZoneRowStatus  RowStatus
}
```

```
Dot20AnSectorCdmaSubSegEntry ::= SEQUENCE
```

```
{
    dot20AnInterlaceId      Integer32,
    dot20AnCdmaSubSegmentNum Integer32,
}
```

```

        dot20AnSectorCdmaSubSegRowStatus RowStatus
    }

Dot20AnSectorConfigEntry ::= SEQUENCE
{
    dot20AnTotalNumSubcarriers      Integer32,
    dot20AnNumGuardSubcarriers      Integer32,
    dot20AnFlSubzoneSize            Integer32,
    dot20AnResourceChannelMuxMode   Integer32,
    dot20AnNumDRCHSubzones          Integer32,
    dot20AnFLReservedInterlaces     INTEGER,
    dot20AnNumFLReservedSubzones    Integer32,
    dot20AnCpichHoppingMode         Integer32,
    dot20AnNumEffectiveAntennas     Integer32,
    dot20AnNumCommonSegmentHopPorts Integer32,
    dot20AnNumLABSegments           Integer32,
    dot20AnMinScchResourceIndex     Integer32,
    dot20AnSinglePAForXCarriers     Integer32,
    dot20AnFlSdmaNumSubtrees        Integer32,
    dot20AnFLDPICHCodeOffsetSubtree0 Integer32,
    dot20AnFLDPICHCodeOffsetSubtree1 Integer32,
    dot20AnFLDPICHCodeOffsetSubtree2 Integer32,
    dot20AnFLDPICHCodeOffsetSubtree3 Integer32,
    dot20AnNumCmnPilotTxAnt         Integer32,
    dot20AnSCCHModSymPerBlock       Integer32,
    dot20AnUseDrchForFlcs           Integer32,
    dot20AnEnableExpandedQPCH       TruthValue,
    dot20AnSectorConfigRowStatus    RowStatus
}

Dot20AnSectorExtChanInfoEntry ::= SEQUENCE
{
    dot20AnPilotPN                  Integer32,
    dot20AnHalfDuplexModeSupported  TruthValue,
    dot20AnFACKBandwidthFactor      Integer32,
    dot20AnFSPBandwidthFactor       Integer32,
    dot20AnSFNCCellID              Integer32,
    dot20AnCellNullID              Integer32,
    dot20AnMaxNumSharedLABs         Integer32,
    dot20AnHalfNumHighOrdScchBl     Integer32,
    dot20AnMaxNumLABs              Integer32,
    dot20AnMax16QamScchBlocks       Integer32,
    dot20AnPdcamResSharingEnabled   TruthValue,
    dot20AnNumAckableLABs           Integer32,
    dot20An16QamScchT2PRatio        INTEGER,
    dot20AnTransmitPower            Integer32,
    dot20AnAssignmentAckHARQTx      Integer32,
    dot20AnFlAuxPilotPower          Integer32,
    dot20AnCommonPilotTransmitPower Integer32,
    dot20AnCDMAInterlacesBitmap     Integer32,
    dot20AnNumOdcchReports          Integer32,
    dot20AnNumRLCdmaSubsegments     Integer32,
    dot20AnNumRLGlobalHoppingZones  Integer32,
    dot20AnRackBandwidthFactor      Integer32,
    dot20AnRlNumSdmaDimensions      Integer32,
    dot20AnRlDpichCodeOffsetSubtree0 Integer32,
    dot20AnRlDpichCodeOffsetSubtree1 Integer32,
    dot20AnRlDpichCodeOffsetSubtree2 Integer32,
    dot20AnRlDpichCodeOffsetSubtree3 Integer32,
    dot20Andot20AnRlSubzoneSize     Integer32,
    dot20AnNumGhbSubzones           Integer32,
    dot20AnSilenceIntervalPeriod     Integer32,
    dot20AnSilenceIntervalOffset     Integer32,
    dot20AnSilenceIntervalDuration   Integer32,
    dot20AnSilenceIntSubzoneMask     Integer32,
    dot20AnAckInterferenceOffset     Integer32,
    dot20AnMacIdRange               INTEGER,
    dot20AnFlPcReportInterval        Integer32,

```

```

dot20AnFlPqiReportInterval      Integer32,
dot20AnFIotReportInterval      Integer32,
dot20AnFastIoTEEnabled         TruthValue,
dot20AnFastOSIEnabled          TruthValue,
dot20AnRabEnabled              TruthValue,
dot20AnOsiResponseMode         INTEGER,
dot20AnSlowInterferenceOffset  Integer32,
dot20AnCtrlAccessOffset       Integer32,
dot20AnRlAuxPilotPower         Integer32,
dot20AnReqQoSPowerBoost        Integer32,
dot20AnErasureTargetCtoI0      Integer32,
dot20AnErasureTargetCtoI1      Integer32,
dot20AnErasureTargetCtoI2      Integer32,
dot20AnErasureTargetCtoI3      Integer32,
dot20AnAccessCycleDuration     Integer32,
dot20AnMaxProbesPerSequence    Integer32,
dot20AnProbeRampUpStepSize     Integer32,
dot20AnPilotThreshold1        Integer32,
dot20AnPilotThreshold2        Integer32,
dot20AnOpenLoopAdjust          Integer32,
dot20AnAccessRetryPersistance0 Integer32,
dot20AnAccessRetryPersistence1 Integer32,
dot20AnAccessRetryPersistence2 Integer32,
dot20AnAccessRetryPersistence3 Integer32,
dot20AnAccessRetryPersistence4 Integer32,
dot20AnAccessRetryPersistence5 Integer32,
dot20AnAccessRetryPersistence6 Integer32,
dot20AnAccessRetryPersistence7 Integer32,
dot20AnSectorExtChanRowStatus  RowStatus
}

Dot20AnSectorGrpResSetsEntry ::= SEQUENCE
{
    dot20AnResourceSetId      Integer32,
    dot20AnResourceSetBitmap  Integer32,
    dot20AnBRCHSubzoneCyclingEnabled TruthValue,
    dot20AnResourceSetSubZoneSpacing Integer32,
    dot20AnNumResourceSubzones Integer32,
    dot20AnResourceSubzoneOffset Integer32,
    dot20AnResourceSetRowStatus RowStatus
}

Dot20AnSectorIpsiEntry ::= SEQUENCE
{
    dot20AnIpsiIndex      Integer32,
    dot20AnSupportedIpsi Integer32,
    dot20AnIpsiRowStatus RowStatus
}

Dot20AnSectorParamEntry ::= SEQUENCE
{
    dot20AnCountryCode      Integer32,
    dot20AnSectorID         OCTET STRING,
    dot20AnChannelBandRef   Integer32,
    dot20AnLatitude         Integer32,
    dot20AnLongitude        Integer32,
    dot20AnLeapSeconds      Integer32,
    dot20AnLocalTimeOffset  Integer32,
    dot20IsSynchronous      TruthValue,
    dot20AnPrimaryRegZoneCode Integer32,
    dot20AnAnGroupId        Integer32,
    dot20AnPilotGroupId     Integer32,
    dot20AnSynchronousGroupId Integer32,
    dot20AnCellGroupId      Integer32,
    dot20AnSectorParamRowStatus RowStatus
}

Dot20AnSectorToIfIndexEntry ::= SEQUENCE

```

```

{
    dot20AnIfChannelBandRef Integer32
}

Dot20CmnAuthStatsEntry ::= SEQUENCE
{
    dot20CmnAuthFailureCounts Counter64,
    dot20CmnAuthSuccessCounts Counter64
}

Dot20CmnLMACPacketStatsEntry ::= SEQUENCE
{
    dot20CmnPacketFormatIndex Integer32,
    dot20CmnARQAttemptsIndex Integer32,
    dot20CmnFwdTxPacketCounts Counter64,
    dot20CmnRevRxPacketCounts Counter64
}

Dot20CmnLMACStatsEntry ::= SEQUENCE
{
    dot20CmnFLABCounts Counter64,
    dot20CmnRLABCounts Counter64,
    dot20CmnAccessGrantCounts Counter64
}

Dot20CmnQmpStatsEntry ::= SEQUENCE
{
    dot20CmnActiveReservationsCounts Counter64,
    dot20CmnIdleReservationsCounts Counter64,
    dot20CmnReservationOpenCounts Counter64,
    dot20CmnReservationCloseCounts Counter64,
    dot20CmnReservationFailCounts Counter64
}

Dot20CmnRlpStatsEntry ::= SEQUENCE
{
    dot20CmnStreamId Integer32,
    dot20CmnRlpTxBytes Counter64,
    dot20CmnRlpReTxBytes Counter64,
    dot20CmnRlpTxDropBytes Counter64,
    dot20CmnRlpTxStatus Counter64,
    dot20CmnRlpRxBytes Counter64,
    dot20CmnRlpRxStatus Counter64,
    dot20CmnRlpTxPackets Counter64,
    dot20CmnRlpReTxPackets Counter64,
    dot20CmnRlpTxrDropPackets Counter64,
    dot20CmnRlpRxPackets Counter64,
    dot20CmnRlpTxNAKTimeouts Counter64,
    dot20CmnRlpTxACKTimeouts Counter64
}

dot20An OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "AN specific configuration and statistics."
    ::= { ieee802dot20 1 }

dot20AnMac OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "MAC layer objects"
    ::= { dot20An 1 }

dot20AnConnectionControl OBJECT IDENTIFIER ::= { dot20AnMac 3 }

dot20AnIdleState OBJECT IDENTIFIER ::= { dot20AnConnectionControl 1 }

dot20AnIdleStateStatsTable OBJECT-TYPE

```



```

SYNTAX          SEQUENCE OF Dot20AnIdleStateStatsEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "This table provides one row of Idle State protocol statistics
    per 802.20 interface (i.e. sector for a specific ChannelBand)
    and carrier."
 ::= { dot20AnIdleState 1 }

dot20AnIdleStateStatsEntry OBJECT-TYPE
SYNTAX          Dot20AnIdleStateStatsEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "An Entry (conceptual row) in the IdleStateStats table. This
    table is indexed by ifIndex and CarrierID. ifIndex: Each IEEE
    802.20 interface (uniquely identified by SectorID) is
    represented by an ifEntry. In the case of a multicarrier
    Sector, the carrierID identifies one specific carrier."
INDEX
    { ifIndex }
 ::= { dot20AnIdleStateStatsTable 1 }

dot20AnAccessAttemptCounts OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of Access Attempts"
 ::= { dot20AnIdleStateStatsEntry 1 }

dot20AnAccessAttemptFailCounts OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of Failed Access Attempts"
 ::= { dot20AnIdleStateStatsEntry 2 }

dot20AnPageAttemptCounts OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of Page Attempts"
 ::= { dot20AnIdleStateStatsEntry 3 }

dot20AnPageFailureCounts OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of Failed Page Attempts"
 ::= { dot20AnIdleStateStatsEntry 4 }

dot20AnOverheadMessages OBJECT IDENTIFIER ::= { dot20AnConnectionControl 4 }

dot20AnSectorConfigTable OBJECT-TYPE
SYNTAX          SEQUENCE OF Dot20AnSectorConfigEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "This table provides one row per 802.20 interface, i.e. sector
    for a specific ChannelBand. This table's attributes specify the
    configuration of the corresponding sector, and can be used to
    populate fields in SystemInfo block and QuickChannelInfo
    message."

```

```

 ::= { dot20AnOverheadMessages 1 }

dot20AnSectorConfigEntry OBJECT-TYPE
    SYNTAX      Dot20AnSectorConfigEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An Entry (conceptual row) in the SectorConfig table. This
         table is indexed by IfIndex. ifIndex: Each IEEE 802.20
         interface (uniquely identified by SectorID) is represented by
         an ifEntry."
    INDEX
        { ifIndex }
    ::= { dot20AnSectorConfigTable 1 }

dot20AnTotalNumSubcarriers OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This parameter shall take the value  $2^{(7+n)}$ , where n is the
         value of the 3 bit field. This field shall not be set to a
         value of 5 or above."
    ::= { dot20AnSectorConfigEntry 29 }

dot20AnNumGuardSubcarriers OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This attribute shall determine the number of guard subcarriers
         as defined in 802.20 Physical layer specification."
    ::= { dot20AnSectorConfigEntry 30 }

dot20AnFlSubzoneSize OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall determine the number of subzones on the
         forward link. If n=0, this parameter shall be set to 64 and if
         n=1, this parameter shall be set to 128."
    ::= { dot20AnSectorConfigEntry 31 }

dot20AnResourceChannelMuxMode OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall determine the number of subzones on the
         forward link. If n=0, this parameter shall be set to 64 and if
         n=1, this parameter shall be set to 128."
    ::= { dot20AnSectorConfigEntry 32 }

dot20AnNumDRCHSubzones OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall take values between 0 and
         NFFT/FLSubzoneSize-1"
    ::= { dot20AnSectorConfigEntry 33 }

dot20AnFLReservedInterlaces OBJECT-TYPE
    SYNTAX      INTEGER {
        zero(1),
        zeroToOne(2),

```

```

        zeroToTwo(3),
        zeroToThree(4),
        zeroToFour(5),
        zeroToFive(6),
        zeroToSix(7),
        zeroToSeven(8),
        zeroAndThree(9),
        zeroAndSix(10),
        zeroTwoAndFour(11),
        zeroTwoFourAndSix(12),
        reserved(13),
        reserved2(14),
        reserved3(15),
        none(16)
    }
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This attribute shall determine which interlaces contain
        reserved bandwidth on the forward link."
    ::= { dot20AnSectorConfigEntry 34 }

dot20AnNumFLReservedSubzones OBJECT-TYPE
    SYNTAX          Integer32 (0..15)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This field determines the number of subzones that are reserved
        on each interlace that contains reserved bandwidth"
    ::= { dot20AnSectorConfigEntry 35 }

dot20AnCpichHoppingMode OBJECT-TYPE
    SYNTAX          Integer32 (0..1)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This field shall be set to 0 for deterministic, and 1 for
        random hopping"
    ::= { dot20AnSectorConfigEntry 36 }

dot20AnNumEffectiveAntennas OBJECT-TYPE
    SYNTAX          Integer32 (1..8)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This attribute shall determine the effective number of
        antennas."
    ::= { dot20AnSectorConfigEntry 37 }

dot20AnNumCommonSegmentHopPorts OBJECT-TYPE
    SYNTAX          Integer32 (0..7)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This attribute shall determine the number of common segment
        hop ports encoded as described in the AIS."
    ::= { dot20AnSectorConfigEntry 38 }

dot20AnNumLABSegments OBJECT-TYPE
    SYNTAX          Integer32 (0..7)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "TThis field shall indicate the number of LABSegments."
    ::= { dot20AnSectorConfigEntry 39 }

dot20AnMinScchResourceIndex OBJECT-TYPE
    SYNTAX          Integer32 (0..31)

```

```

MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall be in units of NFFT/32 resources, and spans
    from 0 to NFFT -1"
::= { dot20AnSectorConfigEntry 40 }

dot20AnSinglePAForXCarriers OBJECT-TYPE
SYNTAX          Integer32 (0..1)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall determine the structure of F-BPICH"
::= { dot20AnSectorConfigEntry 41 }

dot20AnFlSdmaNumSubtrees OBJECT-TYPE
SYNTAX          Integer32 (1..4)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall determine the number of sub-trees on the
    forward link."
::= { dot20AnSectorConfigEntry 42 }

dot20AnFLDPICHCodeOffsetSubtree0 OBJECT-TYPE
SYNTAX          Integer32 (0..3)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall be set to the corresponding value for subtree
    0"
::= { dot20AnSectorConfigEntry 43 }

dot20AnFLDPICHCodeOffsetSubtree1 OBJECT-TYPE
SYNTAX          Integer32 (0..3)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall be set to the corresponding value for subtree
    1"
::= { dot20AnSectorConfigEntry 44 }

dot20AnFLDPICHCodeOffsetSubtree2 OBJECT-TYPE
SYNTAX          Integer32 (0..3)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall be set to the corresponding value for subtree
    2"
::= { dot20AnSectorConfigEntry 45 }

dot20AnFLDPICHCodeOffsetSubtree3 OBJECT-TYPE
SYNTAX          Integer32 (0..3)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall be set to the corresponding value for subtree
    3"
::= { dot20AnSectorConfigEntry 46 }

dot20AnNumCmnPilotTxAnt OBJECT-TYPE
SYNTAX          Integer32 (1..4)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This attribute shall determine the number of common pilot
    transmit antennas."
::= { dot20AnSectorConfigEntry 47 }

```

```

dot20AnSCCHModSymPerBlock OBJECT-TYPE
    SYNTAX      Integer32 (0..3)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall determine the number of modulation symbols
        for each block carried by the F-SCCH"
    ::= { dot20AnSectorConfigEntry 48 }

dot20AnUseDrchForFlcs OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field determines the hopping pattern on the FLCS"
    ::= { dot20AnSectorConfigEntry 49 }

dot20AnEnableExpandedQPCH OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall determine the number of packets delivered to
        the Physical Layer by the MAC Layer"
    ::= { dot20AnSectorConfigEntry 50 }

dot20AnSectorConfigRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the SectorConfig Table. If
        the implementor of this MIB has chosen not to implement
        'dynamic assignment' of sectors, this attribute is not useful
        and should return noSuchName upon SNMP request."
    DEFVAL      { active }
    ::= { dot20AnSectorConfigEntry 78 }

dot20AnSectorExtChanInfoTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnSectorExtChanInfoEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides one row per 802.20 interface, i.e. sector
        for a specific ChannelBand. This table's attributes specify the
        configuration of the corresponding sector, and can be used to
        populate fields in extendedChannelInfo message."
    ::= { dot20AnOverheadMessages 2 }

dot20AnSectorExtChanInfoEntry OBJECT-TYPE
    SYNTAX      Dot20AnSectorExtChanInfoEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An Entry (conceptual row) in the SectorExtChanInfo table. This
        table is indexed by IfIndex. ifIndex: Each IEEE 802.20
        interface (uniquely identified by SectorID) is represented by
        an ifEntry."
    INDEX
        { ifIndex }
    ::= { dot20AnSectorExtChanInfoTable 1 }

dot20AnPilotPN OBJECT-TYPE
    SYNTAX      Integer32 (0..1023)
    MAX-ACCESS   read-write
    STATUS      current

```

```

DESCRIPTION
    "This attribute shall be set to the PilotPN of the sector."
    ::= { dot20AnSectorExtChanInfoEntry 1 }

dot20AnHalfDuplexModeSupported OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This attribute shall be set to True if the access network
        supports half duplex terminals, and shall be set to False
        otherwise. If half-duplex terminals are supported, the access
        network should assign MAC IDs and channel assignments in a
        manner that enables half-duplex terminal operation. A
        half-duplex access terminal is not required to monitor forward
        link transmissions on a PHY frame where it is scheduled to make
        a reverse link transmission."
    ::= { dot20AnSectorExtChanInfoEntry 2 }

dot20AnFACKBandwidthFactor OBJECT-TYPE
    SYNTAX      Integer32 (1..4)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "FACK bandwidth factor"
    ::= { dot20AnSectorExtChanInfoEntry 3 }

dot20AnFSPBBandwidthFactor OBJECT-TYPE
    SYNTAX      Integer32 (1..4)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "FSPB bandwidth factor"
    ::= { dot20AnSectorExtChanInfoEntry 4 }

dot20AnSFNCellID OBJECT-TYPE
    SYNTAX      Integer32 (0..511)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This field determines the Id of the SFNCell"
    ::= { dot20AnSectorExtChanInfoEntry 5 }

dot20AnCellNullID OBJECT-TYPE
    SYNTAX      Integer32 (0..511)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Cell Null Id"
    ::= { dot20AnSectorExtChanInfoEntry 6 }

dot20AnMaxNumSharedLABs OBJECT-TYPE
    SYNTAX      Integer32 (1..4)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This field shall determine the maximum number of shared LABs
        that are transmitted by this sector"
    ::= { dot20AnSectorExtChanInfoEntry 7 }

dot20AnHalfNumHighOrdScchBl OBJECT-TYPE
    SYNTAX      Integer32 (0..3)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This field shall be set to the number of SCCH Blocks that use
        higher order modulation, in units of two"
    ::= { dot20AnSectorExtChanInfoEntry 8 }

```

```

dot20AnMaxNumLABs OBJECT-TYPE
    SYNTAX      Integer32 (0..63)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall be set to the Maximum number of LABs that can
        be transmitted by this sector"
    ::= { dot20AnSectorExtChanInfoEntry 9 }

dot20AnMax16QamScchBlocks OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall be set to the maximum number of 16-QAM blocks
        that may be transmitted by the access network"
    ::= { dot20AnSectorExtChanInfoEntry 10 }

dot20AnPdcamResSharingEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field determines if resource sharing using PDCAMs is
        enabled"
    ::= { dot20AnSectorExtChanInfoEntry 11 }

dot20AnNumAckableLABs OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall be set to the number of LABs on SCCH that the
        access terminal shall acknowledge"
    ::= { dot20AnSectorExtChanInfoEntry 12 }

dot20An16QamScchT2PRatio OBJECT-TYPE
    SYNTAX      INTEGER {
        minusSevenDb(1),
        minusFourDb(2),
        zeroDb(3),
        minusSixteenDb(4)
    }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "16 Qam Scch T2P Ratio"
    ::= { dot20AnSectorExtChanInfoEntry 13 }

dot20AnTransmitPower OBJECT-TYPE
    SYNTAX      Integer32 (0..63)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This attribute shall be set to the transmit power of the
        sector in units of dBm"
    ::= { dot20AnSectorExtChanInfoEntry 14 }

dot20AnAssignmentAckHARQTx OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "The value 0 indicates that no ACK is sent in response to an
        assignment. The rules for interpreting other values of this
        field are provided in the MAC Layer. The value 7 is reserved"
    ::= { dot20AnSectorExtChanInfoEntry 15 }

```

```

dot20AnFlAuxPilotPower OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The attribute's value noted n shall determine the power
        spectral density of the F-AuxPICH relative to the F-ACQCH. The
        pilot power density shall be equal to (-4 + n*0.5) dB."
    ::= { dot20AnSectorExtChanInfoEntry 16 }

dot20AnCommonPilotTransmitPower OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The attribute's value noted n shall determine the power
        spectral density of the F-CPICH during the FL PHY frame
        relative to the F-ACQCH. The pilot power density shall be equal
        to (-4 + n*0.5) dB."
    ::= { dot20AnSectorExtChanInfoEntry 17 }

dot20AnCDMAInterlacesBitmap OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The jth bit of this field shall be set to 1 if interlace i
        contains a Reverse Link CDMA Segment. Here j is assumed to range
        from 0 through 7"
    ::= { dot20AnSectorExtChanInfoEntry 18 }

dot20AnNumOdcchReports OBJECT-TYPE
    SYNTAX      Integer32 (0..31)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "Num ODCCH reports, specified in units of 16"
    ::= { dot20AnSectorExtChanInfoEntry 27 }

dot20AnNumRLCdmaSubsegments OBJECT-TYPE
    SYNTAX      Integer32 (1..32)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall determine the number of RLCdmaSubsegments on
        this sector."
    ::= { dot20AnSectorExtChanInfoEntry 28 }

dot20AnNumRLGlobalHoppingZones OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall be used by the Physical Layer to determine
        the hop pattern for the sector. This parameter shall take the
        value 2^(n+1). Allowed values for this parameter are 2, 4, 8,
        16 and 32"
    ::= { dot20AnSectorExtChanInfoEntry 29 }

dot20AnRackBandwidthFactor OBJECT-TYPE
    SYNTAX      Integer32 (0..3)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This parameter shall be set to 2^n, where n is the value of
        the two bit field."
    ::= { dot20AnSectorExtChanInfoEntry 30 }

```



```

dot20AnRlNumSdmaDimensions OBJECT-TYPE
    SYNTAX      Integer32 (1..4)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall determine the number of spatial dimensions on
        the reverse link."
    ::= { dot20AnSectorExtChanInfoEntry 31 }

dot20AnRlDpichCodeOffsetSubtree0 OBJECT-TYPE
    SYNTAX      Integer32 (0..3)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall be set to the code offset for tree 0"
    ::= { dot20AnSectorExtChanInfoEntry 32 }

dot20AnRlDpichCodeOffsetSubtree1 OBJECT-TYPE
    SYNTAX      Integer32 (0..3)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall be set to the code offset for tree 1"
    ::= { dot20AnSectorExtChanInfoEntry 33 }

dot20AnRlDpichCodeOffsetSubtree2 OBJECT-TYPE
    SYNTAX      Integer32 (0..3)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall be set to the code offset for tree 2"
    ::= { dot20AnSectorExtChanInfoEntry 34 }

dot20AnRlDpichCodeOffsetSubtree3 OBJECT-TYPE
    SYNTAX      Integer32 (0..3)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall be set to the code offset for tree 3"
    ::= { dot20AnSectorExtChanInfoEntry 35 }

dot20Andot20AnRlSubzoneSize OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall determine the size of subzones on the reverse
        link. If n=0, this parameter shall take the value 64 and if
        n=1, this parameter shall take the value 128"
    ::= { dot20AnSectorExtChanInfoEntry 36 }

dot20AnNumGhbSubzones OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall take values between 0 and
        NFFT/RLSubzoneSize-1"
    ::= { dot20AnSectorExtChanInfoEntry 37 }

dot20AnSilenceIntervalPeriod OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall determine the period in units of superframes
        when the silence interval repeats. The SilenceInterval shall

```

```

        take values  $2^n$  superframes, where n is the value of this four
        bit field"
 ::= { dot20AnSectorExtChanInfoEntry 38 }

dot20AnSilenceIntervalOffset OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This specifies the phase of the silence interval in units of 8
        OFDM symbols, i.e., the location of the silence interval within
        the supframe where the silence interval occurs"
 ::= { dot20AnSectorExtChanInfoEntry 39 }

dot20AnSilenceIntervalDuration OBJECT-TYPE
    SYNTAX      Integer32 (1..8)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall determine the duration silence interval in
        units of 8 OFDM symbols"
 ::= { dot20AnSectorExtChanInfoEntry 40 }

dot20AnSilenceIntSubzoneMask OBJECT-TYPE
    SYNTAX      Integer32 (0..255)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field specifies the set of subzones that are blanked
        during the silence interval. If the jth bit of this field is
        equal to one, the jth octile of subzones shall be blanked"
 ::= { dot20AnSectorExtChanInfoEntry 41 }

dot20AnAckInterferenceOffset OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall take values in units of dB"
 ::= { dot20AnSectorExtChanInfoEntry 42 }

dot20AnMacIdRange OBJECT-TYPE
    SYNTAX      INTEGER {
        upTo63(1),
        upTo127(2),
        upTo255(3),
        upTo511(4),
        upTo1023(5),
        upTo2047(6),
        reserved(7),
        upTo31(8)
    }
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This field shall be set to indicate the range of assigned
        MACID values in the sector. For example, a MACIDRange of 63
        indicates that the sector has not assigned MACID values 64 and
        above"
 ::= { dot20AnSectorExtChanInfoEntry 43 }

dot20AnFlPcReportInterval OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS      current

```

DESCRIPTION
 "This field shall determine the periodicity at which power control commands are sent to the access terminal. This parameter shall take the value 2^n , where n is the value of the three bit field."
 ::= { dot20AnSectorExtChanInfoEntry 44 }

dot20AnFlPqiReportInterval OBJECT-TYPE
 SYNTAX Integer32 (0..7)
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "This field shall determine the periodicity at which PQI reports commands are sent by this sector. This parameter shall take the value $16 \cdot 2^n$, where n is the value of the three bit field"
 ::= { dot20AnSectorExtChanInfoEntry 45 }

dot20AnFIotReportInterval OBJECT-TYPE
 SYNTAX Integer32 (0..7)
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "This field shall determine the periodicity at which IoT values are sent to the access terminal. This parameter shall take the value 2^n , where n is the value of the three bit field"
 ::= { dot20AnSectorExtChanInfoEntry 46 }

dot20AnFastIoTEnabled OBJECT-TYPE
 SYNTAX TruthValue
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "This field shall determine if the access terminal is required to read Fast IoT from this sector"
 ::= { dot20AnSectorExtChanInfoEntry 47 }

dot20AnFastOSIEnabled OBJECT-TYPE
 SYNTAX TruthValue
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "This field shall determine if the access terminal is required to read OSI from this sector"
 ::= { dot20AnSectorExtChanInfoEntry 48 }

dot20AnRabEnabled OBJECT-TYPE
 SYNTAX TruthValue
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "This field shall be set to 1 if this sector transmits RAB, and shall be set to 1 otherwise"
 ::= { dot20AnSectorExtChanInfoEntry 49 }

dot20AnOsiResponseMode OBJECT-TYPE
 SYNTAX INTEGER {
 stochastic(1),
 deterministic(2)
 }
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "This field shall determine the type of response to OSI modes"
 ::= { dot20AnSectorExtChanInfoEntry 50 }

dot20AnSlowInterferenceOffset OBJECT-TYPE
SYNTAX Integer32 (0..15)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This field shall be set in units of dB"
::= { dot20AnSectorExtChanInfoEntry 51 }

dot20AnCtrlAccessOffset OBJECT-TYPE
SYNTAX Integer32 (0..7)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This field determines the initial gain of the R-CQICH over the R-ACH. The value of this parameter shall be $-11+n$ dB, where n is the value of this field"
::= { dot20AnSectorExtChanInfoEntry 52 }

dot20AnRlAuxPilotPower OBJECT-TYPE
SYNTAX Integer32 (0..7)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This field shall be determine the offset of R-AuxPICH with respect to R-PICH. This parameter shall take the value $4+n$."
::= { dot20AnSectorExtChanInfoEntry 53 }

dot20AnReqQoSPowerBoost OBJECT-TYPE
SYNTAX Integer32 (0..3)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This field shall be in units of dB"
::= { dot20AnSectorExtChanInfoEntry 54 }

dot20AnErasureTargetCtoI0 OBJECT-TYPE
SYNTAX Integer32 (0..15)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This attribute's value noted n shall determine the transmit power of erasure sequences for different assignment sizes. The transmit power shall be equal to $n-6$ dB."
::= { dot20AnSectorExtChanInfoEntry 55 }

dot20AnErasureTargetCtoI1 OBJECT-TYPE
SYNTAX Integer32 (0..15)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This attribute's value noted n shall determine the transmit power of erasure sequences for different assignment sizes. The transmit power shall be equal to $n-6$ dB."
::= { dot20AnSectorExtChanInfoEntry 56 }

dot20AnErasureTargetCtoI2 OBJECT-TYPE
SYNTAX Integer32 (0..15)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This attribute's value noted n shall determine the transmit power of erasure sequences for different assignment sizes. The transmit power shall be equal to $n-6$ dB."
::= { dot20AnSectorExtChanInfoEntry 57 }

dot20AnErasureTargetCtoI3 OBJECT-TYPE
SYNTAX Integer32 (0..15)
MAX-ACCESS read-write

```

STATUS          current
DESCRIPTION
    "This attribute's value noted n shall determine the transmit
    power of erasure sequences for different assignment sizes. The
    transmit power shall be equal to n-6 dB."
::= { dot20AnSectorExtChanInfoEntry 58 }

dot20AnAccessCycleDuration OBJECT-TYPE
SYNTAX          Integer32 (1..4)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This attribute shall determine the duration of the access
    cycle in units of Control Segments (as defined by the 802.20
    AIS spec Physical Layer)."
::= { dot20AnSectorExtChanInfoEntry 59 }

dot20AnMaxProbesPerSequence OBJECT-TYPE
SYNTAX          Integer32 (1..16)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This attribute shall determine the maximum number of probe
    sequences that can be part of one access sequence."
::= { dot20AnSectorExtChanInfoEntry 60 }

dot20AnProbeRampUpStepSize OBJECT-TYPE
SYNTAX          Integer32 (0..15)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This attribute's value noted n shall determine the power ramp
    up used for probes within a probe sequence and shall indicate a
    ramp up value of 0.5*(1+n) dB."
::= { dot20AnSectorExtChanInfoEntry 61 }

dot20AnPilotThreshold1 OBJECT-TYPE
SYNTAX          Integer32 (0..7)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This attribute's value noted n shall determine PilotThreshold1
    used by the Access Channel MAC Protocol. The value shall be -2n
    dB."
::= { dot20AnSectorExtChanInfoEntry 62 }

dot20AnPilotThreshold2 OBJECT-TYPE
SYNTAX          Integer32 (0..7)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This attribute's value noted n shall determine PilotThreshold2
    used by the Access Channel MAC Protocol. The value shall be -2n
    dB."
::= { dot20AnSectorExtChanInfoEntry 63 }

dot20AnOpenLoopAdjust OBJECT-TYPE
SYNTAX          Integer32 (0..255)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This attribute's value noted n shall determine the nominal
    power to be used by access terminal in the open loop power
    estimate. The value of nominal power shall be 70+n dB."
::= { dot20AnSectorExtChanInfoEntry 64 }

dot20AnAccessRetryPersistence0 OBJECT-TYPE
SYNTAX          Integer32 (0..7)

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION

"This attribute shall determine the persistence probability for determining access sequence backoff. If this attribute's value is set to n, the access terminal shall use 2^{-n} as the retry

persistence."
 ::= { dot20AnSectorExtChanInfoEntry 65 }

dot20AnAccessRetryPersistence1 OBJECT-TYPE

SYNTAX Integer32 (0..7)
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"This attribute shall determine the persistence probability for determining access sequence backoff. If this attribute's value is set to n, the access terminal shall use 2^{-n} as the retry persistence."

::= { dot20AnSectorExtChanInfoEntry 66 }

dot20AnAccessRetryPersistence2 OBJECT-TYPE

SYNTAX Integer32 (0..7)
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"This attribute shall determine the persistence probability for determining access sequence backoff. If this attribute's value is set to n, the access terminal shall use 2^{-n} as the retry persistence."

::= { dot20AnSectorExtChanInfoEntry 67 }

dot20AnAccessRetryPersistence3 OBJECT-TYPE

SYNTAX Integer32 (0..7)
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"This attribute shall determine the persistence probability for determining access sequence backoff. If this attribute's value is set to n, the access terminal shall use 2^{-n} as the retry persistence."

::= { dot20AnSectorExtChanInfoEntry 68 }

dot20AnAccessRetryPersistence4 OBJECT-TYPE

SYNTAX Integer32 (0..7)
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"This attribute shall determine the persistence probability for determining access sequence backoff. If this attribute's value is set to n, the access terminal shall use 2^{-n} as the retry persistence."

::= { dot20AnSectorExtChanInfoEntry 69 }

dot20AnAccessRetryPersistence5 OBJECT-TYPE

SYNTAX Integer32 (0..7)
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"This attribute shall determine the persistence probability for determining access sequence backoff. If this attribute's value is set to n, the access terminal shall use 2^{-n} as the retry

```

        persistence."
 ::= { dot20AnSectorExtChanInfoEntry 70 }

dot20AnAccessRetryPersistence6 OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This attribute shall determine the persistence probability for
        determining access sequence backoff. If this attribute's value
        is set to n, the access terminal shall use 2^-n as the retry
        persistence."
 ::= { dot20AnSectorExtChanInfoEntry 71 }

dot20AnAccessRetryPersistence7 OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This attribute shall determine the persistence probability for
        determining access sequence backoff. If this attribute's value
        is set to n, the access terminal shall use 2^-n as the retry
        persistence."
 ::= { dot20AnSectorExtChanInfoEntry 72 }

dot20AnSectorExtChanRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the SectorExtChanInfo
        Table. If the implementor of this MIB has chosen not to
        implement 'dynamic assignment' of sectors, this attribute is
        not useful and should return noSuchName upon SNMP request."
    DEFVAL      { active }
 ::= { dot20AnSectorExtChanInfoEntry 73 }

dot20AnSectorParamTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnSectorParamEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table provides one row per 802.20 carrier of a sector for
        a specific ChannelBand. This table's attributes specify the
        configuration of the corresponding sector and can be used to
        populate fields in Sthe SectorParameters message."
 ::= { dot20AnOverheadMessages 3 }

dot20AnSectorParamEntry OBJECT-TYPE
    SYNTAX      Dot20AnSectorParamEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An Entry (conceptual row) in the SectorParam table. This table
        is indexed by ifIndex. ifIndex: Each IEEE 802.20 interface
        (uniquely identified by SectorID) is represented by an
        ifEntry."
    INDEX
        { ifIndex }
 ::= { dot20AnSectorParamTable 1 }

dot20AnCountryCode OBJECT-TYPE
    SYNTAX      Integer32 (0..4096)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION

```

```

        "This attribute shall be set to the three digit Mobile Country
        Code associated with this sector (as specified in ITU-T
        Recommendation E.212, Identification Plan for Land Mobile
        Stations)."
```

::= { dot20AnSectorParamEntry 1 }

dot20AnSectorID OBJECT-TYPE

```

    SYNTAX      OCTET STRING (SIZE(16))
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "Sector Address Identifier. The access network shall set the
        value of the SectorID according to the rules specified in IEEE
        802.20 AIS. The access terminal shall not assume anything about
        the format of the SectorID other than that it uniquely
        identifies the sector."
```

::= { dot20AnSectorParamEntry 2 }

dot20AnChannelBandRef OBJECT-TYPE

```

    SYNTAX      Integer32
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "The reference to the ChannelBand defined in ChannelBands table
        using this value as index (dot20AnChannelBandIndex)"
```

::= { dot20AnSectorParamEntry 3 }

dot20AnLatitude OBJECT-TYPE

```

    SYNTAX      Integer32 (-1296000..1296000)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "The latitude of the sector. This attribute shall be set to
        this sector's latitude in units of 0.25 second, expressed as a
        two's complement signed number with positive numbers signifying
        North latitudes. This attribute shall be set to a value in the
        range 1296000 to 1296000 inclusive (corresponding to a range of
        -90 to +90)."
```

::= { dot20AnSectorParamEntry 4 }

dot20AnLongitude OBJECT-TYPE

```

    SYNTAX      Integer32 (-2592000..2592000)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "The longitude of the sector. This attribute shall be set to
        this sector's longitude in units of 0.25 second, expressed as a
        two's complement signed number with positive numbers signifying
        East longitude. This attribute shall be set to a value in the
        range 2592000 to 2592000 inclusive (corresponding to a range of
        -180° to +180°)."
```

::= { dot20AnSectorParamEntry 5 }

dot20AnLeapSeconds OBJECT-TYPE

```

    SYNTAX      Integer32 (0..255)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "The number of leap seconds that have occurred since the start
        of system time."
```

::= { dot20AnSectorParamEntry 6 }

dot20AnLocalTimeOffset OBJECT-TYPE

```

    SYNTAX      Integer32 (0..2047)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This attribute shall be set to the offset of the local time
```



```

        from System Time. This value will be in units of minutes,
        expressed as a two's complement signed number."
 ::= { dot20AnSectorParamEntry 7 }

dot20IsSynchronous OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This attribute shall be set to true if all sectors in the
        deployment are synchronous. This attribute shall be set to
        false otherwise."
 ::= { dot20AnSectorParamEntry 8 }

dot20AnPrimaryRegZoneCode OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "The PrimaryRegistrationZoneCode for this sector"
 ::= { dot20AnSectorParamEntry 9 }

dot20AnAnGroupId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "Sector's AN Group Id"
 ::= { dot20AnSectorParamEntry 10 }

dot20AnPilotGroupId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "Sector's Pilot Group Id"
 ::= { dot20AnSectorParamEntry 11 }

dot20AnSynchronousGroupId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "Sector's Synchronous Group Id"
 ::= { dot20AnSectorParamEntry 12 }

dot20AnCellGroupId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "Sector's Cell Group Id"
 ::= { dot20AnSectorParamEntry 13 }

dot20AnSectorParamRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the SectorParam Table. If
        the implementor of this MIB has chosen not to implement
        'dynamic assignment' of sectors, this attribute is not useful
        and should return noSuchName upon SNMP request."
    DEFVAL      { active }
 ::= { dot20AnSectorParamEntry 14 }

dot20AnSectorGrpResSetsTable OBJECT-TYPE

```

```

SYNTAX          SEQUENCE OF Dot20AnSectorGrpResSetsEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "This table provides one row per 802.20 sector and Forward
    Channel group resource set (see ExtendedChannelInfo message in
    AIS)."
```

::= { dot20AnOverheadMessages 4 }

dot20AnSectorGrpResSetsEntry OBJECT-TYPE

```

SYNTAX          Dot20AnSectorGrpResSetsEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "An Entry (conceptual row) in the
    AnSectorFwdChanGrpResourceSets table. This table is indexed by
    ifIndex and resourceSetId ifIndex: Each IEEE 802.20 interface
    (uniquely identified by SectorID) is represented by an
    ifEntry."
```

INDEX

```

{ ifIndex, dot20AnResourceSetId }
::= { dot20AnSectorGrpResSetsTable 1 }
```

dot20AnResourceSetId OBJECT-TYPE

```

SYNTAX          Integer32 (0..7)
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "Index of the forward channel group resource set for a
    particular sector."
```

::= { dot20AnSectorGrpResSetsEntry 1 }

dot20AnResourceSetBitmap OBJECT-TYPE

```

SYNTAX          Integer32 (0..255)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "The jth bit of this field shall be set to 1 if a frame with
    frame index mod InterlaceDepth = j contains a subzone that
    corresponds to this resource set. If the InterlaceDepth = 6,
    the last two bits of this field shall be set to 0"
```

::= { dot20AnSectorGrpResSetsEntry 2 }

dot20AnBRCHSubzoneCyclingEnabled OBJECT-TYPE

```

SYNTAX          TruthValue
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall be set to 1 if BRCHSubzoneCycling is enabled
    on this sector. For BRCH resource set with BRCHSubzoneCycling
    disabled or DRCH resource set, the first subzone offset on all
    interlaces where this resource set is present shall be set to
    the ResourceSubzoneOffset. For BRCH resource set with
    BRCHSubzoneCycling enabled, the offset of the first subzone
    over each interlace is shifted cyclically. Since the offset of
    first subzone over the lowest indexed interlace is defined by
    ResourceSubzoneOffset, the offset of the first subzone in the
    next interlace, where the resource set is present, is increased
    by 1 mod NumBRCHSubzones"
```

::= { dot20AnSectorGrpResSetsEntry 3 }

dot20AnResourceSetSubZoneSpacing OBJECT-TYPE

```

SYNTAX          Integer32 (0..3)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This field shall indicate the spacing between subzones in a
```

```

        resource set. Subzones belonging to a resource group on an
        interlace shall be equally spaced, where the first subzone is
        defined by ResourceSubzoneOffset and
        BRCHSubzoneCyclingEnabled"
 ::= { dot20AnSectorGrpResSetsEntry 4 }

dot20AnNumResourceSubzones OBJECT-TYPE
    SYNTAX      Integer32 (0..31)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall determine the number of subzones in each
        interlace where the resource set is present. An interlace is
        defined as the set of frames that have the same Frame Index mod
        InterlaceDepth, where InterlaceDepth is defined by
        ResourceSetInterlace. This parameter takes the value n+1."
 ::= { dot20AnSectorGrpResSetsEntry 5 }

dot20AnResourceSubzoneOffset OBJECT-TYPE
    SYNTAX      Integer32 (0..31)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This field shall be set to the first subzone on the lowest
        indexed interlace that is part of a resource set. Interlace
        index i is defined for the set of frames that have Frame Index
        mod InterlaceDepth = i, where InterlaceDepth is defined by
        ResourceSetInterlace"
 ::= { dot20AnSectorGrpResSetsEntry 6 }

dot20AnResourceSetRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the
        SectorFwdChanGrpResourceSet Table. If the implementor of this
        MIB has chosen not to implement 'dynamic assignment' of
        sectors, this attribute is not useful and should return
        noSuchName upon SNMP request."
    DEFVAL      { active }
 ::= { dot20AnSectorGrpResSetsEntry 7 }

dot20AnSecondaryRegZoneCodeTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnSecondaryRegZoneCodeEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table provides one row per 802.20 interface and per
        secondary registration zone code."
 ::= { dot20AnOverheadMessages 5 }

dot20AnSecondaryRegZoneCodeEntry OBJECT-TYPE
    SYNTAX      Dot20AnSecondaryRegZoneCodeEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An Entry (conceptual row) in the SecondaryRegZoneCode table.
        This table is indexed by IfIndex and
        dot20AnSecondaryRegZoneCodeIndex. ifIndex: Each IEEE 802.20
        interface (uniquely identified by SectorID) is represented by
        an ifEntry."
    INDEX
        { ifIndex, dot20AnSecondaryRegZoneCodeIndex }
 ::= { dot20AnSecondaryRegZoneCodeTable 1 }

```

```

dot20AnSecondaryRegZoneCodeIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Index of the secondary registration zone code for a particular
        sector."
    ::= { dot20AnSecondaryRegZoneCodeEntry 1 }

dot20AnSecRegZoneCode OBJECT-TYPE
    SYNTAX      Integer32 (0..255)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "One of the SecondaryRegistrationZoneCode for this sector"
    ::= { dot20AnSecondaryRegZoneCodeEntry 2 }

dot20AnSecondaryRegZoneRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the SecondaryRegZoneCode
        Table. If the implementor of this MIB has chosen not to
        implement 'dynamic assignment' of sectors, this attribute is
        not useful and should return noSuchName upon SNMP request."
    DEFVAL      { active }
    ::= { dot20AnSecondaryRegZoneCodeEntry 3 }

dot20AnSectorIpsiTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnSectorIpsiEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table provides one row per 802.20 interface and per
        Ipsi."
    ::= { dot20AnOverheadMessages 6 }

dot20AnSectorIpsiEntry OBJECT-TYPE
    SYNTAX      Dot20AnSectorIpsiEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An Entry (conceptual row) in the SectorIpsi table. This table
        is indexed by IfIndex and dot20AnIpsiIndex. ifIndex: Each IEEE
        802.20 interface (uniquely identified by SectorID) is
        represented by an ifEntry."
    INDEX
        { ifIndex }
    ::= { dot20AnSectorIpsiTable 1 }

dot20AnIpsiIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Index of an Ipsi supported by a particular sector."
    ::= { dot20AnSectorIpsiEntry 1 }

dot20AnSupportedIpsi OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "IPSI supported by a particular sector"
    ::= { dot20AnSectorIpsiEntry 2 }

```

```

dot20AnIpsiRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the SectorIpsi Table. If
        the implementor of this MIB has chosen not to implement
        'dynamic assignment' of sectors, this attribute is not useful
        and should return noSuchName upon SNMP request."
    DEFVAL       { active }
    ::= { dot20AnSectorIpsiEntry 3 }

dot20AnSectorCdmaSubSegTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnSectorCdmaSubSegEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table provides one row per 802.20 sector, interlace and
        Reverse Channel group CDMA Sub segment (see ExtendedChannelInfo
        message in AIS)."
    ::= { dot20AnOverheadMessages 8 }

dot20AnSectorCdmaSubSegEntry OBJECT-TYPE
    SYNTAX      Dot20AnSectorCdmaSubSegEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An Entry (conceptual row) in the AnSectorCdmaSubSeg table.
        This table is indexed by ifIndex, interlaceId and
        CDMASubSegmentId."
    INDEX
        { ifIndex, dot20AnInterlaceId }
    ::= { dot20AnSectorCdmaSubSegTable 1 }

dot20AnInterlaceId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Interlace Id"
    ::= { dot20AnSectorCdmaSubSegEntry 1 }

dot20AnCdmaSubSegmentNum OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "Number of reverse channel CDMA Sub segment within an interlace
        for a particular sector."
    ::= { dot20AnSectorCdmaSubSegEntry 2 }

dot20AnSectorCdmaSubSegRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the SectorCdmaSubSeg
        Table. If the implementor of this MIB has chosen not to
        implement 'dynamic assignment' of sectors, this attribute is
        not useful and should return noSuchName upon SNMP request."
    DEFVAL       { active }
    ::= { dot20AnSectorCdmaSubSegEntry 3 }

dot20AnChannelBandsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnChannelBandsEntry

```

```
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "This table provides one row per 802.20 ChannelBand. This
    table's attributes specify the ChannelBand record of a
    particular ChannelBand which may be used for a sector defined
    in the SectorConfig table, or as a neighbor to one sector
    defined in the SectorConfig table."
 ::= { dot20AnOverheadMessages 9 }

dot20AnChannelBandsEntry OBJECT-TYPE
SYNTAX        Dot20AnChannelBandsEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "An Entry (conceptual row) in the ChannelBands table. This
    table is indexed by ChannelBandIndex."
INDEX
    { dot20AnChannelBandIndex }
 ::= { dot20AnChannelBandsTable 1 }

dot20AnChannelBandIndex OBJECT-TYPE
SYNTAX        Integer32 (1..2147483647)
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "Index of the ChannelBand within the ChannelBands table."
 ::= { dot20AnChannelBandsEntry 1 }

dot20AnChannelBandClass OBJECT-TYPE
SYNTAX        Integer32 (0..31)
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    "This attribute shall be set to the band class number
    corresponding to the frequency assignment of the ChannelBand
    specified by this record."
 ::= { dot20AnChannelBandsEntry 2 }

dot20AnChannelNumber OBJECT-TYPE
SYNTAX        Integer32 (0..2047)
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    "This attribute shall be set to the Channel number
    corresponding to the frequency assignment of the ChannelBand
    specified by this record."
 ::= { dot20AnChannelBandsEntry 3 }

dot20AnReverseChannelBandClass OBJECT-TYPE
SYNTAX        Integer32 (0..31)
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    "This attribute shall be set to the band class number
    corresponding to the frequency assignment of the reverse
    ChannelBand specified by this record."
 ::= { dot20AnChannelBandsEntry 4 }

dot20AnReverseChannelNumber OBJECT-TYPE
SYNTAX        Integer32 (0..2047)
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    "This attribute shall be set to the Channel number
    corresponding to the frequency assignment of the Reverse
    ChannelBand specified by this record."
 ::= { dot20AnChannelBandsEntry 5 }
```

```

dot20AnChannelBandShortId OBJECT-TYPE
    SYNTAX      Integer32 (0..3)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This attribute identifies the two bit index that identifies
        this channel in beacon transmissions"
    ::= { dot20AnChannelBandsEntry 6 }

dot20AnChannelBandAccessHashMask OBJECT-TYPE
    SYNTAX      Integer32 (0..65536)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "this attribute is set to the AccessHashingChannelMask for this
        channel"
    ::= { dot20AnChannelBandsEntry 7 }

dot20AnChannelBandStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the ChannelBands Table.
        If the implementor of this MIB has chosen not to implement
        'dynamic assignment' of ChannelBands, this attribute is not
        useful and should return noSuchName upon SNMP request."
    DEFVAL      { active }
    ::= { dot20AnChannelBandsEntry 8 }

dot20AnNeighborSectorsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnNeighborSectorsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides one row per 802.20 neighbor sector. This
        table's attributes specify the sector parameters of a
        particular neighbor sector which may be used as a neighbor to
        one sector defined in the SectorConfig table."
    ::= { dot20AnOverheadMessages 10 }

dot20AnNeighborSectorsEntry OBJECT-TYPE
    SYNTAX      Dot20AnNeighborSectorsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An Entry (conceptual row) in the AnNeighborSectors table. This
        table is indexed by ChannelBandIndex, NeighborSectorIndex."
    INDEX
        { dot20AnChannelBandIndex, dot20AnNeighborSectorIndex }
    ::= { dot20AnNeighborSectorsTable 1 }

dot20AnNeighborSectorIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "Index of the Neighbor Sector for this Neighbor Carrier within
        the ChannelBand."
    ::= { dot20AnNeighborSectorsEntry 1 }

dot20AnNeighborSectorID OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(16))
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION

```

```

        "Sector Address Identifier. The access network shall set the
        value of the SectorID according to the rules specified in IEEE
        802.20 AIS. The access terminal shall not assume anything about
        the format of the SectorID other than that it uniquely
        identifies the sector."
    ::= { dot20AnNeighborSectorsEntry 2 }

dot20AnNeighborPilotPN OBJECT-TYPE
    SYNTAX      Integer32 (0..4095)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This attribute shall be set to the PilotPN of a neighboring
        sector that the access terminal should add to its Neighbor
        Set."
    ::= { dot20AnNeighborSectorsEntry 3 }

dot20AnNeighborTransmitPower OBJECT-TYPE
    SYNTAX      Integer32 (0..63)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This attribute shall be set to the transmit power of the
        sector in units of dBm."
    ::= { dot20AnNeighborSectorsEntry 4 }

dot20AnNeighborGloballySynch OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This attribute shall be set to true if the sector transmitting
        this pilot is synchronous to system time."
    ::= { dot20AnNeighborSectorsEntry 5 }

dot20AnNeighborSectorAnGroupId OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "Neighbor Sector's AN Group Id"
    ::= { dot20AnNeighborSectorsEntry 7 }

dot20AnNeighborSectorPilotGrpId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "Neighbor Sector's Pilot Group Id"
    ::= { dot20AnNeighborSectorsEntry 8 }

dot20AnNeighborSynchGroupId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "Neighbor Sector's Synchronous Group Id"
    ::= { dot20AnNeighborSectorsEntry 9 }

dot20AnNeighborSectorCellGroupId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "Neighbor Sector's Cell Group Id"
    ::= { dot20AnNeighborSectorsEntry 10 }

```



```

dot20AnNeighborSectorStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the NeighborSectors
        Table. If the implementor of this MIB has chosen not to
        implement 'dynamic assignment' of neighbor sectors this
        attribute is not useful and should return noSuchName upon SNMP
        request."
    DEFVAL      { active }
    ::= { dot20AnNeighborSectorsEntry 11 }

dot20AnOtherTechNghbrsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnOtherTechNghbrsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides one row per other technology neighbor
        channel. This table's attributes specify the technology type
        and neighborlist of a particular neighbor channel which may be
        used by one sector defined in the SectorConfig table for
        inter-technology handoff."
    ::= { dot20AnOverheadMessages 11 }

dot20AnOtherTechNghbrsEntry OBJECT-TYPE
    SYNTAX      Dot20AnOtherTechNghbrsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An Entry (conceptual row) in the AnOtherTechNghbrs table. This
        table is indexed by Sector (ifIndex) and OtherTechnologyIndex"
    INDEX
        { ifIndex, dot20AnOtherTechnologyIndex }
    ::= { dot20AnOtherTechNghbrsTable 1 }

dot20AnOtherTechnologyIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "The neighbor other technology entry index"
    ::= { dot20AnOtherTechNghbrsEntry 1 }

dot20AnTechnologyType OBJECT-TYPE
    SYNTAX      Integer32 (0..255)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This attribute shall be set to the type of other technology.
        Interpretation for its value should as defined in the AIS
        spec."
    ::= { dot20AnOtherTechNghbrsEntry 2 }

dot20AnTechNghbrListLength OBJECT-TYPE
    SYNTAX      Integer32 (0..255)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "This attribute shall be set the length, in bytes, of the
        neighbor list information for the other technology."
    ::= { dot20AnOtherTechNghbrsEntry 3 }

dot20AnTechnologyNeighborList OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(256))
    MAX-ACCESS   read-write
    STATUS      current

```

```

DESCRIPTION
    "This attribute shall be set to the neighbor list information
    for the other technology."
 ::= { dot20AnOtherTechNghbrsEntry 4 }

dot20AnOtherTechNghbrRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the OtherTechNghbrs Table.
        If the implementor of this MIB has chosen not to implement
        'dynamic assignment' of other technology neighbors, this
        attribute is not useful and should return noSuchName upon SNMP
        request."
    DEFVAL      { active }
    ::= { dot20AnOtherTechNghbrsEntry 5 }

dot20AnNeighborListTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnNeighborListEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table defines the neighbor lists for the sectors defined
        in the SectorConfig table. Each row in this table indexed per
        sector (ifIndex) specifies a pointer to a neighbor sector of
        this sector."
    ::= { dot20AnOverheadMessages 12 }

dot20AnNeighborListEntry OBJECT-TYPE
    SYNTAX      Dot20AnNeighborListEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An Entry (conceptual row) in the AnNeighborList table. This
        table is indexed by Sector (ifIndex) and NeighborIndex indexing
        each neighbor sector for a particular Sector."
    INDEX
        { ifIndex, dot20AnNeighborIndex }
    ::= { dot20AnNeighborListTable 1 }

dot20AnNeighborIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..32)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This index identifies one neighbor sector for a Sector."
    ::= { dot20AnNeighborListEntry 1 }

dot20AnNeighborSectorPointer OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This attribute points to an instance of sector in SectorConfig
        table or in NeighborSectors table. This sector is defined as a
        neighbor of the sector identified by the ifIndex of this
        attribute's entry."
    ::= { dot20AnNeighborListEntry 2 }

dot20AnNeighborRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
        instances of the columnar objects in the NeighborList Table.

```

If the implementor of this MIB has chosen not to implement 'dynamic assignment' of neighbor list entries this attribute is not useful and should return noSuchName upon SNMP request."

```

DEFVAL      { active }
 ::= { dot20AnNeighborListEntry 3 }

dot20AnSectorToIfIndexTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20AnSectorToIfIndexEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table can be used to find the ifIndex of an 802.20
        interface based on its SectorID and ChannelBand information
        (reverse mapping of the Sector Config table)."
    ::= { dot20An 2 }

dot20AnSectorToIfIndexEntry OBJECT-TYPE
    SYNTAX      Dot20AnSectorToIfIndexEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An Entry (conceptual row) in the AnSectorToIfIndex table."
    INDEX
        { dot20AnSectorID, ifIndex }
    ::= { dot20AnSectorToIfIndexTable 1 }

dot20AnIfChannelBandRef OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The reference to the ChannelBand defined in ChannelBands table
        (dot20AnChannelBandIndex)"
    ::= { dot20AnSectorToIfIndexEntry 1 }

dot20Cmn OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "Common configuration and statistics."
    ::= { ieee802dot20 2 }

dot20CmnMac OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION
        "MAC layer objects"
    ::= { dot20Cmn 1 }

dot20CmnSessionControl OBJECT IDENTIFIER ::= { dot20CmnMac 1 }

dot20CmnSessionMgtProtocol OBJECT IDENTIFIER ::= { dot20CmnSessionControl 1 }

dot20CmnSessionOpenCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of session opened"
    ::= { dot20CmnSessionMgtProtocol 1 }

dot20CmnSessionCloseCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of session closed"
    ::= { dot20CmnSessionMgtProtocol 2 }

```

```
dot20CmnSessionFailureCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of session open/close failures"
    ::= { dot20CmnSessionMgtProtocol 3 }

dot20CmnConnectionControl OBJECT IDENTIFIER ::= { dot20CmnMac 3 }

dot20CmnConnectedState OBJECT IDENTIFIER ::= { dot20CmnConnectionControl 1 }

dot20CmnActiveConnectionCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of current active connections"
    ::= { dot20CmnConnectedState 1 }

dot20CmnConnectionAttemptCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of connection attempts"
    ::= { dot20CmnConnectedState 2 }

dot20CmnConnectionFailureCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of connection failures during connection attempt."
    ::= { dot20CmnConnectedState 3 }

dot20CmnConnectionDropCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of dropped connection after a connection has been
        established."
    ::= { dot20CmnConnectedState 4 }

dot20CmnConnectionReleaseCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of connection release after a connection has been
        established."
    ::= { dot20CmnConnectedState 5 }

dot20CmnRadioLink OBJECT IDENTIFIER ::= { dot20CmnMac 4 }

dot20CmnRlp OBJECT IDENTIFIER ::= { dot20CmnRadioLink 2 }

dot20CmnRlpStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20CmnRlpStatsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides one row of Radio Link Protocol statistics
        per 802.20 interface"
    ::= { dot20CmnRlp 1 }

dot20CmnRlpStatsEntry OBJECT-TYPE
```

```

SYNTAX      Dot20CmnRlpStatsEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An Entry (conceptual row) in the RlpStats table. This table is
    indexed by IfIndex and dot20StreamId."
INDEX
    { ifIndex, dot20CmnStreamId }
::= { dot20CmnRlpStatsTable 1 }

dot20CmnStreamId OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Stream Id"
::= { dot20CmnRlpStatsEntry 1 }

dot20CmnRlpTxBytes OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of RLP bytes of payload transmitted"
::= { dot20CmnRlpStatsEntry 2 }

dot20CmnRlpReTxBytes OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of RLP bytes of payload retransmitted"
::= { dot20CmnRlpStatsEntry 3 }

dot20CmnRlpTxDropBytes OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of RLP bytes of dropped before transmission"
::= { dot20CmnRlpStatsEntry 4 }

dot20CmnRlpTxStatus OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of RLP ReceiverStatus messages transmitted"
::= { dot20CmnRlpStatsEntry 5 }

dot20CmnRlpRxBytes OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of RLP bytes of payload received"
::= { dot20CmnRlpStatsEntry 6 }

dot20CmnRlpRxStatus OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of RLP ReceiverStatus messages received"
::= { dot20CmnRlpStatsEntry 7 }

dot20CmnRlpTxPackets OBJECT-TYPE

```

```
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of RLP Packets transmitted"
 ::= { dot20CmnRlpStatsEntry 8 }

dot20CmnRlpReTxPackets OBJECT-TYPE
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of RLP Packets retransmitted"
 ::= { dot20CmnRlpStatsEntry 9 }

dot20CmnRlpTxrDropPackets OBJECT-TYPE
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of RLP Packets dropped before transmission"
 ::= { dot20CmnRlpStatsEntry 10 }

dot20CmnRlpRxPackets OBJECT-TYPE
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of RLP Packets received"
 ::= { dot20CmnRlpStatsEntry 11 }

dot20CmnRlpTxNAKTimeouts OBJECT-TYPE
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of NAK Timeouts"
 ::= { dot20CmnRlpStatsEntry 12 }

dot20CmnRlpTxACKTimeouts OBJECT-TYPE
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Number of ACK Timeouts"
 ::= { dot20CmnRlpStatsEntry 13 }

dot20CmnQmp OBJECT-IDENTITY
STATUS          current
DESCRIPTION
    "Qos Management Protocol"
 ::= { dot20CmnRadioLink 3 }

dot20CmnQmpStatsTable OBJECT-TYPE
SYNTAX          SEQUENCE OF Dot20CmnQmpStatsEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "This table provides one row of QMP statistics per 802.20
     interface"
 ::= { dot20CmnQmp 2 }

dot20CmnQmpStatsEntry OBJECT-TYPE
SYNTAX          Dot20CmnQmpStatsEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "An Entry (conceptual row) in the QmpStats table. This table is
```

```

        indexed by IfIndex. ifIndex: Each IEEE 802.20 interface is
        represented by an ifEntry."
INDEX
    { ifIndex }
 ::= { dot20CmnQmpStatsTable 1 }

dot20CmnActiveReservationsCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of Active Reservations"
    ::= { dot20CmnQmpStatsEntry 1 }

dot20CmnIdleReservationsCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of Idle Reservations"
    ::= { dot20CmnQmpStatsEntry 2 }

dot20CmnReservationOpenCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of Reservations Open requests"
    ::= { dot20CmnQmpStatsEntry 3 }

dot20CmnReservationCloseCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of Reservations Close requests"
    ::= { dot20CmnQmpStatsEntry 4 }

dot20CmnReservationFailCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of Failed Reservations requests"
    ::= { dot20CmnQmpStatsEntry 5 }

dot20CmnSecurity OBJECT IDENTIFIER ::= { dot20CmnMac 5 }

dot20CmnKeyExchangeProtocol OBJECT IDENTIFIER ::= { dot20CmnSecurity 1 }

dot20CmnKeyExchangeAttemptCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of key exchanges attempts"
    ::= { dot20CmnKeyExchangeProtocol 1 }

dot20CmnKeyExchangeFailureCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of key exchanges failures"
    ::= { dot20CmnKeyExchangeProtocol 2 }

dot20CmnMessageIntegrityProtocol OBJECT IDENTIFIER ::= { dot20CmnSecurity 2
}

```

```

dot20CmnAuthStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20CmnAuthStatsEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table provides one row of Authentication statistics per
        802.20 interface (i.e. sector for a specific ChannelBand)."
```

::= { dot20CmnMessageIntegrityProtocol 1 }

```

dot20CmnAuthStatsEntry OBJECT-TYPE
    SYNTAX      Dot20CmnAuthStatsEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Authentication statistics per 802.20 interfaces"
```

INDEX

{ ifIndex }

::= { dot20CmnAuthStatsTable 1 }

```

dot20CmnAuthFailureCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of Authentication failures"
```

::= { dot20CmnAuthStatsEntry 1 }

```

dot20CmnAuthSuccessCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Number of successful Authentications"
```

::= { dot20CmnAuthStatsEntry 2 }

```

dot20CmnLowerMAC OBJECT IDENTIFIER ::= { dot20CmnMac 6 }
```

```

dot20CmnLMACPacketStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20CmnLMACPacketStatsEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table provides one row of Lower MAC protocol statistics
        per 802.20 interface, packet format and nb of ARQ attempts
        needed in order to successfully transmit/receive a packet."
```

::= { dot20CmnLowerMAC 1 }

```

dot20CmnLMACPacketStatsEntry OBJECT-TYPE
    SYNTAX      Dot20CmnLMACPacketStatsEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An Entry (conceptual row) in the LMACPacketStats table. This
        table is indexed by IfIndex, PacketFormatIndex and
        ARQAttemptsIndex."
```

INDEX

{ ifIndex, dot20CmnPacketFormatIndex, dot20CmnARQAttemptsIndex }

::= { dot20CmnLMACPacketStatsTable 1 }

```

dot20CmnPacketFormatIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The packet format index as defined in 802.20 AIS spec."
```

::= { dot20CmnLMACPacketStatsEntry 1 }


```

dot20CmnARQAttemptsIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "Number of ARQ attempts that were needed in order to transmit
         or receive a packet. Index 0 means that the packets failed to
         be transmitted/received."
    ::= { dot20CmnLMACPacketStatsEntry 2 }

dot20CmnFwdTxPacketCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of transmitted packets"
    ::= { dot20CmnLMACPacketStatsEntry 3 }

dot20CmnRevRxPacketCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of received packets"
    ::= { dot20CmnLMACPacketStatsEntry 4 }

dot20CmnLMACStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot20CmnLMACStatsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides one row of Lower MAC protocol statistics
         per 802.20 interface and packet formats."
    ::= { dot20CmnLowerMAC 2 }

dot20CmnLMACStatsEntry OBJECT-TYPE
    SYNTAX      Dot20CmnLMACStatsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An Entry (conceptual row) in the LMACStats table. This table
         is indexed by IfIndex, PacketFormatIndex."
    INDEX
        { ifIndex, dot20CmnPacketFormatIndex }
    ::= { dot20CmnLMACStatsTable 1 }

dot20CmnFLABCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of Forward Link Assignment Blocks"
    ::= { dot20CmnLMACStatsEntry 1 }

dot20CmnRLABCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of Reverse Link Assignment Block"
    ::= { dot20CmnLMACStatsEntry 2 }

dot20CmnAccessGrantCounts OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION

```

```

        "Number of Access Grants"
        ::= { dot20CmnLMACStatsEntry 3 }

dot20Conformance OBJECT IDENTIFIER ::= { ieee802dot20 4 }

dot20Groups OBJECT IDENTIFIER ::= { dot20Conformance 1 }

dot20CmnSessionMgtPGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnSessionCloseCounts, dot20CmnSessionFailureCounts,
          dot20CmnSessionOpenCounts }
    STATUS      current
    DESCRIPTION
        "The session management protocol statistics"
    ::= { dot20Groups 1 }

dot20CmnKeyExchangePGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnKeyExchangeAttemptCounts,
          dot20CmnKeyExchangeFailureCounts }
    STATUS      current
    DESCRIPTION
        "The key exchange protocol statistics"
    ::= { dot20Groups 4 }

dot20CmnConnectedStatePGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnActiveConnectionCounts,
          dot20CmnConnectionAttemptCounts, dot20CmnConnectionDropCounts,
          dot20CmnConnectionFailureCounts, dot20CmnConnectionReleaseCounts
        }
    STATUS      current
    DESCRIPTION
        "The connected state protocol statistics"
    ::= { dot20Groups 5 }

dot20CmnRadioLinkGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnActiveReservationsCounts,
          dot20CmnIdleReservationsCounts, dot20CmnReservationCloseCounts,
          dot20CmnReservationFailCounts, dot20CmnReservationOpenCounts,
          dot20CmnRevRxPacketCounts, dot20CmnRlpReTxBytes,
          dot20CmnRlpReTxPackets, dot20CmnRlpRxBytes,
          dot20CmnRlpRxPackets, dot20CmnRlpRxStatus,
          dot20CmnRlpTxACKTimeouts, dot20CmnRlpTxBytes,
          dot20CmnRlpTxDropBytes, dot20CmnRlpTxNAKTimeouts,
          dot20CmnRlpTxPackets, dot20CmnRlpTxStatus,
          dot20CmnRlpTxrDropPackets }
    STATUS      current
    DESCRIPTION
        "The radio link layer statistics"
    ::= { dot20Groups 7 }

dot20CmnAuthGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnAuthFailureCounts, dot20CmnAuthSuccessCounts }
    STATUS      current
    DESCRIPTION
        "The authentication protocol statistics"
    ::= { dot20Groups 8 }

dot20CmnLowerMACGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnAccessGrantCounts, dot20CmnFLABCounts,
          dot20CmnFwdTxPacketCounts, dot20CmnRLABCounts,
          dot20CmnRevRxPacketCounts }
    STATUS      current

```

```

DESCRIPTION
    "The lower mac sublayer statistics"
    ::= { dot20Groups 9 }

dot20AnIdleStatePGroup OBJECT-GROUP
    OBJECTS
        { dot20AnAccessAttemptCounts, dot20AnAccessAttemptFailCounts,
          dot20AnPageAttemptCounts, dot20AnPageFailureCounts }
    STATUS
        current
    DESCRIPTION
        "The An idle state protocol statistics"
        ::= { dot20Groups 10 }

dot20AnOverheadGroup OBJECT-GROUP
    OBJECTS
        { dot20An16QamScchT2PRatio, dot20AnAccessCycleDuration,
          dot20AnAccessRetryPersistence0, dot20AnAccessRetryPersistence1,
          dot20AnAccessRetryPersistence2, dot20AnAccessRetryPersistence3,
          dot20AnAccessRetryPersistence4, dot20AnAccessRetryPersistence5,
          dot20AnAccessRetryPersistence6, dot20AnAccessRetryPersistence7,
          dot20AnAckInterferenceOffset, dot20AnAnGroupId,
          dot20AnAssignmentAckHARQTx, dot20AnBRCHSubzoneCyclingEnabled,
          dot20AnCDMAInterlacesBitmap, dot20AnCdmaSubSegmentNum,
          dot20AnCellGroupId, dot20AnCellNullID,
          dot20AnChannelBandAccessHashMask, dot20AnChannelBandClass,
          dot20AnChannelBandRef, dot20AnChannelBandShortId,
          dot20AnChannelNumber, dot20AnCommonPilotTransmitPower,
          dot20AnCountryCode, dot20AnCpichHoppingMode,
          dot20AnCtrlAccessOffset, dot20AnEnableExpandedQPCH,
          dot20AnErasureTargetCtoI0, dot20AnErasureTargetCtoI1,
          dot20AnErasureTargetCtoI2, dot20AnErasureTargetCtoI3,
          dot20AnFACKBandwidthFactor, dot20AnFIotReportInterval,
          dot20AnFLDPICHCodeOffsetSubtree0,
          dot20AnFLDPICHCodeOffsetSubtree1,
          dot20AnFLDPICHCodeOffsetSubtree2,
          dot20AnFLDPICHCodeOffsetSubtree3, dot20AnFLReservedInterlaces,
          dot20AnFSPBandwidthFactor, dot20AnFastIoTEnabled,
          dot20AnFastOSIEnabled, dot20AnFlAuxPilotPower,
          dot20AnFlPcReportInterval, dot20AnFlPqiReportInterval,
          dot20AnFlSdmaNumSubtrees, dot20AnFlSubzoneSize,
          dot20AnHalfDuplexModeSupported, dot20AnHalfNumHighOrdScchBl,
          dot20AnIfChannelBandRef, dot20AnLatitude, dot20AnLeapSeconds,
          dot20AnLocalTimeOffset, dot20AnLongitude, dot20AnMacIdRange,
          dot20AnMax16QamScchBlocks, dot20AnMaxNumLABs,
          dot20AnMaxNumSharedLABs, dot20AnMaxProbesPerSequence,
          dot20AnMinScchResourceIndex, dot20AnNeighborGloballySynch,
          dot20AnNeighborPilotPN, dot20AnNeighborSectorAnGroupId,
          dot20AnNeighborSectorCellGroupId, dot20AnNeighborSectorID,
          dot20AnNeighborSectorPilotGrpId, dot20AnNeighborSectorPointer,
          dot20AnNeighborSynchGroupId, dot20AnNeighborTransmitPower,
          dot20AnNumAckableLABs, dot20AnNumCmnPilotTxAnt,
          dot20AnNumCommonSegmentHopPorts, dot20AnNumDRCHSubzones,
          dot20AnNumEffectiveAntennas, dot20AnNumFLReservedSubzones,
          dot20AnNumGhbSubzones, dot20AnNumGuardSubcarriers,
          dot20AnNumLABSegments, dot20AnNumOdcchReports,
          dot20AnNumRLCdmaSubsegments, dot20AnNumRLGlobalHoppingZones,
          dot20AnNumResourceSubzones, dot20AnOpenLoopAdjust,
          dot20AnOsiResponseMode, dot20AnPdcamResSharingEnabled,
          dot20AnPilotGroupId, dot20AnPilotPN, dot20AnPilotThreshold1,
          dot20AnPilotThreshold2, dot20AnPrimaryRegZoneCode,
          dot20AnProbeRampUpStepSize, dot20AnRabEnabled,
          dot20AnRackBandwidthFactor, dot20AnReqQoSPowerBoost,
          dot20AnResourceChannelMuxMode, dot20AnResourceSetBitmap,
          dot20AnResourceSetSubZoneSpacing, dot20AnResourceSubzoneOffset,
          dot20AnReverseChannelBandClass, dot20AnReverseChannelNumber,
          dot20AnRlAuxPilotPower, dot20AnRlDpichCodeOffsetSubtree0,
          dot20AnRlDpichCodeOffsetSubtree1,
          dot20AnRlDpichCodeOffsetSubtree2,

```

```

dot20AnRlDpichCodeOffsetSubtree3, dot20AnRlNumSdmaDimensions,
dot20AnSCCHModSymPerBlock, dot20AnSFNCellID,
dot20AnSecRegZoneCode, dot20AnSectorID,
dot20AnSilenceIntSubzoneMask, dot20AnSilenceIntervalDuration,
dot20AnSilenceIntervalOffset, dot20AnSilenceIntervalPeriod,
dot20AnSinglePForXCarriers, dot20AnSlowInterferenceOffset,
dot20AnSupportedIpsi, dot20AnSynchronousGroupId,
dot20AnTechNghbrListLength, dot20AnTechnologyNeighborList,
dot20AnTechnologyType, dot20AnTotalNumSubcarriers,
dot20AnTransmitPower, dot20AnUseDrchForFlcs,
dot20Andot20AnRlSubzoneSize, dot20IsSynchronous }

```

STATUS current

DESCRIPTION

"The overhead messages protocol configuration"

::= { dot20Groups 11 }

dot20AnOverheadGroup2 OBJECT-GROUP

OBJECTS

```

{ dot20AnChannelBandStatus, dot20AnIpsiRowStatus,
dot20AnNeighborRowStatus, dot20AnNeighborSectorStatus,
dot20AnOtherTechNghbrRowStatus, dot20AnResourceSetRowStatus,
dot20AnSecondaryRegZoneRowStatus,
dot20AnSectorCdmaSubSegRowStatus, dot20AnSectorConfigRowStatus,
dot20AnSectorExtChanRowStatus, dot20AnSectorParamRowStatus }

```

STATUS current

DESCRIPTION

"The overhead messages protocol configuration This group should be implemented if assignment of 802.20 interfaces is to be preformed through snmp."

::= { dot20Groups 12 }

dot20Compliances OBJECT IDENTIFIER ::= { dot20Conformance 2 }

dot20AnCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for SNMPv2 entities that implement the IEEE 802.20 MIB for the An."

MODULE IEEE802dot20-MIB

MANDATORY-GROUPS

```

{ dot20AnIdleStatePGroup, dot20AnOverheadGroup,
dot20CmnAuthGroup, dot20CmnConnectedStatePGroup,
dot20CmnKeyExchangePGroup, dot20CmnLowerMACGroup,
dot20CmnRadioLinkGroup, dot20CmnSessionMgtPGroup }

```

GROUP dot20AnOverheadGroup2

DESCRIPTION

"This group is required only if 'dynamic assignment' of rows in the OverheadGroup tables is supported."

::= { dot20Compliances 1 }

END

18. 625k-MC spectral layout terminology and requirements

The 625k-MC spectral layout terminology and requirements shall be as defined in Clause 2 of ATIS-0700004.2005.

19. 625k-MC slot and frame structure

The 625k-MC mode MAC frame and slot structures for each “carrier” whose RF channel bandwidth is 625 kHz, are specified in this clause. The RF carrier frequency in the carrier allocation shall be a consecutive set of frequencies separated by 625 kHz.

19.1 Overview

An overview of the 625k-MC Frame and Slot structure is the same as that presented in 3.1 of ATIS-0700004.2005.

19.2 RF channel and frame structure

As defined in 3.2 of ATIS-0700004.2005.

19.3 Burst formats

19.3.1 Frequency synchronization

As defined in 3.3.1 of ATIS-0700004.2005.

19.3.2 Timing synchronization

As defined in 3.3.2 of ATIS-0700004.2005.

19.3.3 Broadcast burst

As defined in 3.3.3 of ATIS-0700004.2005.

19.3.4 Page burst

As defined in 3.3.4 of ATIS-0700004.2005.

19.3.5 Configuration request burst

As defined in 3.3.5 of ATIS-0700004.2005.

19.3.6 Standard Uplink Burst (FACCH, RACH, and TCH)

As defined in 3.3.6 of ATIS-0700004.2005.

19.3.7 Standard Downlink Burst (CM, SMB, AA, and TCH)

The Standard Downlink Burst structure and formats (CM, AA, and TCH) shall be as defined in 3.3.7 of ATIS-0700004.2005.

The 625k-MC Short Message Broadcast (SMB) burst shall adhere to the Standard Downlink Burst Format.

20. 625k-MC modulation and channel coding

20.1 625k-MC modulation and channel coding overview

Clause 4 of ATIS-0700004.2005 gives the baseline capabilities for 625k-MC modulation and channel coding. The capability therein described is further enhanced by the requirements of this subclause as underlined below:

625k-MC supports 11 Modulation Classes (ModClasses). ModClasses 0-8 for the downlink and ModClasses 0-7 for uplink shall be as specified in the 4.2 of ATIS-0700004.2005. Table 433 shows the data throughputs for ModClasses 9-10 for the downlink and 8-10 for the uplink, in addition to those presented in Table 21 of ATIS-0700004.2005.

Table 433—User data throughput at various ModClasses

ModClass	Single stream downlink throughput (kbps)	Aggregated 3 stream downlink throughput (kbps)	Single stream uplink throughput (kbps)	Aggregated 3 stream uplink throughput (kbps)
0	35.2	105.6	6.4	19.2
1	49.6	148.8	12.8	38.4
2	81.6	244.8	25.6	76.8
3	126.4	379.2	43.2	129.6
4	161.6	484.8	57.6	172.8
5	198.4	595.2	72.0	216.0
6	262.4	787.2	97.6	292.8
7	307.2	921.6	115.2	345.6
8	353.6	1060.8	<u>132.8</u>	<u>398.4</u>
9	<u>377.6</u>	<u>1132.8</u>	<u>142.4</u>	<u>427.2</u>
10	<u>497.6</u>	<u>1492.8</u>	<u>190.4</u>	<u>571.2</u>

20.2 Standard modulation and coding

The standard modulation and coding for 625k-MC shall be as specified in 4.2 of ATIS-0700004.2005 *with the additional text and tables as underlined below*.

The 625k-MC physical layer supports different data rates by selecting among various coding and modulation schemes. Figure 178 illustrates the coded modulation system that achieves rates from approximately $\frac{1}{2}$ to 5.5 bits/symbol. The notation $\{i, j\}$ indicates that a block is active only for modulation classes i and j . The modulation and signal sets together with the associated parameters for puncturing, shaping and block coding for 625k-MC ModClasses 0-8 shall be as defined in Table 22 of ATIS-0700004.2005. The signal sets together with the associated parameters for puncturing, shaping and block coding for ModClasses 9 and 10 shall be as shown in Table 434. The allowed modulation class (ModClass) values for various burst types shall be as shown in Table 435.



The operation of each block in Figure 178 is detailed in the subsequent subclauses. In the descriptions that follow, $\{a_1, a_2, \dots, a_{N_a}\}$ denotes the sequence of N_a bits input to the encrypt block, $\{b_1, b_2, \dots, b_{N_b}\}$ denotes the N_b bits input to the CRC block. The block sizes $\{N_a, N_b, \dots, N_q\}$ are listed in Table 436 and Table 437. For ModClasses 0-8 in the downlink direction and ModClasses 0-7 in the uplink direction (as a function of the message type, modulation class, and uplink/downlink direction), block sizes $\{N_a, N_b, \dots, N_q\}$ shall be as listed in Table 24 and Table 25, respectively, of ATIS-0700004.2005. For ModClasses 9-10 in the downlink direction and ModClasses 8-10 in the uplink direction, block sizes $\{N_a, N_b, \dots, N_q\}$ shall be as listed in Table 436 and Table 437, respectively. For ModClasses 0-8, the block sizes for CR, CM, RA and AA shall be as shown in Table 26 to 29 of ATIS-0700004.2005. The block sizes for SMB shall be as shown in Table 438.

ModClass	Bits/Sym	Signal Set	Puncture	Shaper	Block Code
0	0.5	$\pi/2$ BPSK	Repeat	—	—
1	0.67	$\pi/2$ BPSK	1 of 4	—	—
2	1	QPSK	—	—	—
3	1.5	QPSK	2 of 6	—	—
4	2	8-PSK	—	—	(64,57)
5	2.5	8-PSK	—	—	(64,57)
6	3	12-QAM	2 of 6	3/4	(48,47)
7	3.5	16-QAM	2 of 6	4/4	(64,63)
8	4	24-QAM	2 of 6	5/4	(80,79)
<u>9</u>	<u>4.5</u>	<u>32-QAM</u>	<u>2 of 6</u>	<u>5/5</u>	<u>(80,79)</u>
<u>10</u>	<u>5.5</u>	<u>64-QAM</u>	<u>2 of 5</u>	<u>6/6</u>	<u>(80,79)</u>
<u>11–15</u>	<u>RESERVED</u>				

Table 435—ModClass versus Burst Type

Logical channel	Burst Type	ModClass
<u>TCH-uplink</u>	<u>Standard uplink</u>	<u>0–10</u>
<u>TCH-downlink</u>	<u>Standard downlink</u>	<u>0–10</u>
RACH-uplink	Standard uplink	0
RACH-downlink	Standard downlink	0
CCH-uplink	Configuration Request	0
CCH-downlink	Standard downlink	0
BCH and PCH	Broadcast and Page	See 4.3 Broadcast channel Modulation and Coding on page 4.22 of ATIS-0700004.2005

Table 436—Block Lengths in Downlink Traffic Burst

ModClass	Na,b	Nc	Nd	Ne	Nf	Ng	Nh	Ni	Nq	N _m	Nm,n,o,p
0	206	16	222	230	460	460	0	0	0	460	460
1	282	16	298	306	613	460	0	0	0	460	460
2	436	16	452	460	920	920	0	0	0	920	460
3	666	16	682	690	1380	920	0	0	0	920	460
4	840	16	452	460	920	920	404	460	460	460	460
5	1021	16	222	230	460	460	815	920	920	460	460
6	1341	16	682	690	1380	920	675	690	920	920	460
7	1571	16	682	690	1380	920	905	920	920	920	460
8	1801	16	682	690	1380	920	1135	1150	920	920	460
<u>9</u>	<u>1919</u>	<u>16</u>	<u>1027</u>	<u>1035</u>	<u>2070</u>	<u>1380</u>	<u>908</u>	<u>920</u>	<u>920</u>	<u>920</u>	<u>460</u>
<u>10</u>	<u>2523</u>	<u>16</u>	<u>950</u>	<u>958</u>	<u>1916</u>	<u>1150</u>	<u>1589</u>	<u>1610</u>	<u>1610</u>	<u>920</u>	<u>460</u>

Table 437—Block Lengths in Uplink Traffic Burst

ModClass	Na,b	Nc	Nd	Ne	Nf	Ng	Nh	Ni	Nq	N _m	Nm,n,o,p
0	67	16	83	91	182	182	0	0	0	182	182
1	97	16	113	121	242	182	0	0	0	182	182

Table 437—Block Lengths in Uplink Traffic Burst (continued)

ModClass	Na,b	Nc	Nd	Ne	Nf	Ng	Nh	Ni	Nq	$\tilde{N}_m$	Nm,n,o,p
2	158	16	174	182	364	364	0	0	0	364	182
3	249	16	265	273	546	364	0	0	0	364	182
4	319	16	174	182	364	364	161	182	182	182	182
5	389	16	83	91	182	182	322	364	364	182	182
6	516	16	265	273	546	364	267	273	364	364	182
7	607	16	265	273	546	364	358	364	364	364	182
8	<u>698</u>	<u>16</u>	<u>265</u>	<u>273</u>	<u>546</u>	<u>364</u>	<u>449</u>	<u>455</u>	<u>364</u>	<u>364</u>	<u>182</u>
9	<u>744</u>	<u>16</u>	<u>401</u>	<u>409</u>	<u>819</u>	<u>546</u>	<u>359</u>	<u>364</u>	<u>364</u>	<u>364</u>	<u>182</u>
<u>10</u>	<u>984</u>	<u>16</u>	<u>371</u>	<u>379</u>	<u>758</u>	<u>455</u>	<u>629</u>	<u>637</u>	<u>637</u>	<u>364</u>	<u>182</u>

Table 438—Block Lengths in Short Message Broadcast

<u>ModClass</u>	<u>Na,b</u>	<u>Nc</u>	<u>Nd</u>	<u>Ne</u>	<u>Nf</u>	<u>Ng</u>	<u>Nh</u>	<u>Ni</u>	<u>Nq</u>	<u>$\tilde{N}_m$</u>	<u>Nm,n,o,p</u>
<u>0</u>	<u>105</u>	<u>16</u>	<u>121</u>	<u>129</u>	<u>258</u>	<u>460</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>460</u>	<u>460</u>

20.2.1 Encryption

As defined in 4.2.1 of ATIS-0700004.2005.

20.2.2 Cyclic redundancy check

As defined in 4.2.2 of ATIS-0700004.2005.

20.2.3 Multiplexing

As defined in 4.2.3 of ATIS-0700004.2005.

20.2.4 Tail append

As defined in 4.2.4 of ATIS-0700004.2005.

20.2.5 Convolutional encoding

As defined in 4.2.5 of ATIS-0700004.2005.

20.2.6 Puncturing and repeating

The puncture and repeat block as shown in the Figure 178, punctures or repeats the coded output bits $f = \{f_1, f_2, \dots\}$ outputs the sequence $g = \{g_1, g_2, \dots\}$.

Puncturing and repeating for ModClasses 0-8 shall be as specified in 4.2.6 of ATIS-0700004.2005. Puncturing and repeating for ModClasses 9-10 shall be specified as:

- For ModClass 9, the coded outputs are punctured by a periodic puncturing pattern that deletes two bits from every block of six as that for ModClass 3, 6, 7 and 8.
- For ModClass 10, the coded outputs are punctured by a periodic puncturing pattern that deletes two bits from every block of five. The Table 1 shows the mapping of coded out put sequence to punctured sequence: The 2nd and 5th bit in each 5bit-block of continuous input bit sequence $f = \{f_1, f_2, \dots\}$ are punctured.

Table 439—Puncture and Repeat Block operation for ModClass 10

f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
g_1		g_2	g_3		g_4		g_5	g_6		g_7		g_8	g_9		g_{10}		g_{11}	g_{12}	

20.2.7 Block coding

As defined in 4.2.7 of ATIS-0700004.2005.

20.2.7.1 Extended Hamming code

As defined in 4.2.7.1 of ATIS-0700004.2005.

20.2.7.2 Parity check code

The parity check code for ModClasses 6, 7, and 8 shall be as defined in 4.2.7.2 of ATIS-0700004.2005.

The parity check code for ModClasses 9 and 10 shall be as that specified in 4.2.7.2 of ATIS-0700004.2005 for ModClass 8 with input block lengths B equal to 79.

20.2.8 Block shaper

ModClasses 6–10 shall employ block shaping. The block shaper shall processes a binary-valued input sequence into a ternary-valued output sequence. The block shaping for ModClasses 6, 7, and 8 shall be as defined in 4.2.8 of ATIS-0700004.2005. For other modulation classes, no block shaping is performed and the output of the block shaper is equal to its input: $q_k = i_k$ for $k = 1, 2, \dots, N_i$.

The input sequence shall be divided into blocks of size B=5 and 6 bits for ModClasses 9 and 10, respectively. ModClasses 9 and 10 shall use 5/5 and 6/6 block shaper as shown in Table 440 and Table 441, respectively.

Table 440—Mapping for Rate 5/5 block shaper

$i_{1+5}, i_{2+5}, i_{3+5}, i_{4+5}, i_{5+5}$	$q_{1+5}, q_{2+5}, q_{3+5}, q_{4+5}, q_{5+5}$
x_1, x_2, x_3, x_4, x_5	x_1, x_2, x_3, x_4, x_5

Table 441—Mapping for Rate 6/6 block shaper

$i_{1+6}, i_{2+6}, i_{3+6}, i_{4+6}, i_{5+6}, i_{6+6}$	$q_{1+6}, q_{2+6}, q_{3+6}, q_{4+6}, q_{5+6}, q_{6+6}$

Table 441—Mapping for Rate 6/6 block shaper

$\tilde{x}_1$ $\tilde{x}_2$ $\tilde{x}_3$ $\tilde{x}_4$ $\tilde{x}_5$ $\tilde{x}_6$	$\tilde{x}_1$ $\tilde{x}_2$ $\tilde{x}_3$ $\tilde{x}_4$ $\tilde{x}_5$ $\tilde{x}_6$

20.2.9 Symbol mapping

Symbol mapping for ModClasses 2, 3, 6, 7, and 8 shall be as defined in 4.2.9 of ATIS-0700004.2005.

For ModClasses 9 and 10, pairs of outputs from the lookup tables, Table 252 and Table 253, shall be multiplexed into a single complex output symbol as specified in 4.2.9 of ATIS-0700004.2005. The symbol mapper for ModClass 9 shall be as shown in Table 442.

Table 442—Symbol mapper for ModClass 9

q_{2k-1}	q_{2k}	g_{3k-2}	g_{3k-1}	g_{3k}	$\tilde{m}$	
					$2k-1$	$2k$
0	0	0	0	0	1	1
0	1	0	0	0	1	-3
1	0	0	0	0	-3	1
1	1	0	0	0	-3	-3
0	0	0	0	1	3	1
0	1	0	0	1	3	-3
1	0	0	0	1	-1	1
1	1	0	0	1	-1	-3
0	0	0	1	0	1	5
0	1	0	1	0	3	-5
1	0	0	1	0	-3	5
1	1	0	1	0	-1	-5
0	0	0	1	1	5	1
0	1	0	1	1	5	-3
1	0	0	1	1	-5	1
1	1	0	1	1	-5	-3
0	0	1	0	0	1	3
0	1	1	0	0	1	-1
1	0	1	0	0	-3	3
1	1	1	0	0	-3	-1
0	0	1	0	1	3	3
0	1	1	0	1	3	-1
1	0	1	0	1	-1	3

Table 442—Symbol mapper for ModClass 9 (continued)

q_{2k-1}	q_{2k}	g_{3k-2}	g_{3k-1}	g_{3k}	$\tilde{m}$	
					$2k-1$	$2k$
1	1	1	0	1	-1	-1
0	0	1	1	0	3	5
0	1	1	1	0	1	-5
1	0	1	1	0	-1	5
1	1	1	1	0	-3	-5
0	0	1	1	1	5	3
0	1	1	1	1	5	-1
1	0	1	1	1	-5	3
1	1	1	1	1	-5	-1

For ModClass10, the mapping rules shall be

$$q_{7k-6} \quad q_{7k-5} \quad g_{5k-4} \rightarrow X_{12k-11} \quad X_{12k-10} \quad X_{12k-9}$$

$$q_{7k-4} \quad q_{7k-3} \quad g_{5k-3} \rightarrow X_{12k-8} \quad X_{12k-7} \quad X_{12k-6}$$

$$q_{7k-2} \quad q_{7k-1} \quad g_{5k-2} \rightarrow X_{12k-5} \quad X_{12k-4} \quad X_{12k-3}$$

$$q_{7k} \quad g_{5k-1} \quad g_{5k} \rightarrow X_{12k-2} \quad X_{12k-1} \quad X_{12k}$$

for $k = 1, 2, \dots$

Table 443—Symbol mapper for ModClass 10

$X_{3l-2} X_{3l-1} X_{3l}$	$\tilde{m}$
0 0 0	-5
0 0 1	-7
0 1 1	-3
0 1 0	-1
1 1 0	3
1 1 1	1
1 0 1	5
1 0 0	7

20.2.10 Interleaving

As defined in 4.2.10 of ATIS-0700004.2005.

20.2.11 Scrambling

As defined in 4.2.11 of ATIS-0700004.2005.

Scrambling for SMB shall be same as that for CM.

20.2.12 $\pi/2$ Rotation and Scaling

Shall be as defined in 4.2.12 of ATIS-0700004.2005.

The modulation scaling for ModClasses 9 and 10 shall be as specified in Table 444.

Table 444—Modulation scaling

ModClass	λ
2	$1/\sqrt{2}$
3	$1/\sqrt{2}$
4	1
5	1
6	$1/\sqrt{7}$
7	$1/\sqrt{10}$
8	$1/\sqrt{14}$
9	$\frac{1}{\sqrt{20}}$
10	$\frac{1}{\sqrt{42}}$

20.3 Broadcast channel modulation and coding

The broadcast logical channels shall be as specified in 4.3 of the ATIS-0700004.2005.

21. 625k-MC user terminal radio transmission and reception

The UT shall follow all procedures as specified in Clause 5 of ATIS-0700004.2005 with supplemental definitions listed herein to support higher modulation formats: 24-QAM, 32-QAM, and 64-QAM and the corresponding ModClasses 9 and 10. Regarding support of modulation classes, for radio compliance per 5.3.5.2 in ATIS-0700004.2005, the UT shall support at least ModClass 0. However, a UT is expected to support additional modulation classes, as indicated in the UT Capability Message. The terminology used for UTs radio compliance specifications is defined in 5.1.1 of ATIS-0700004.2005.

21.1 625k-MC user terminal nominal output power for modulation formats

625k-MC mode UT shall support Nominal output power as defined in Table 41 of ATIS-0700004.2005 and Nominal UT transmit power per carrier for modulation formats 24-QAM, 32-QAM, and 64-QAM shall be as defined in Table 445.

Table 445—Nominal UT transmit power per carrier for 24-QAM, 32-QAM, and 64-QAM

Modulation format	Nominal output power		
	Power class 1	Power class 2	Power class 3
64-QAM	29 dBm	24 dBm	19 dBm
32-QAM	29 dBm	24 dBm	19 dBm
24-QAM	29 dBm	24 dBm	19 dBm

21.2 625k-MC UT modulation accuracy for modulation formats

This subclause defines the performance of the transmit modulator of 625k-MC UT.

The modulation accuracy of the 625k-MC UT transmitter shall be in accordance with the

specifications Table 42 of ATIS-0700004.2005 and specifications in Table 446, which define modulation accuracy requirement for the remaining modulation formats: 24-QAM, 32-QAM, and 64-QAM.

Table 446—UT modulation accuracy modulation formats 24-QAM, 32-QAM, and 64-QAM

Modulation format	Modulation accuracy
64-QAM	< 4%
32-QAM	< 5.5%
24-QAM	< 6%

21.3 625k-MC UT reference sensitivity level

Reference Sensitivity Level of 625k-MC UT radio shall be as defined in Table 447 for ModClass 9 and 10 and shall be as defined in Table 44 of ATIS-0700004.2005 for ModClasses 0 to 8.

Table 447—UT reference sensitivity at FER = 10^{-2} (downlink)

Mod Class	Nominal reference sensitivity (dBm)
9	−90.2
10	−87.2

21.4 625k-MC UT receiver SINR estimation accuracy

Acceptable SINR estimation accuracy Report Values for the 625k-MC UT Receiver shall be as defined in Table 48 of ATIS-0700004.2005 for all ModClasses except ModClasses 9 and 10, whose acceptable SINR value shall be as defined in Table 448.

Table 448—Tabulated range of acceptable SINR Report Values for ModClass 9 and 10 at UT

Input SINR	5 th percentile	95 th percentile (dB)
$S < -3$ dB	no minimum	0 dB
$-3 \leq S < 28$ dB	$S - 3$ dB	$S + 3$ dB
$S \geq 28$ dB	25 dB	$S + 3$ dB

21.5 625k-MC UT receiver adjacent channel selectivity

Adjacent Channel Selectivity of 625k-MC UT radio receiver shall be as defined in Table 449 for ModClasses 9 and 10. For modulation classes 0 to 8, Adjacent Channel Selectivity of 625k-MC UT radio receiver shall be as defined in Table 45 of ATIS-0700004.2005.

Table 449—625k-MC UT adjacent channel selectivity for ModClasses 9 and 10

Interfering carrier	Desired signal modulation class	ACS
First adjacent carrier	9 or 10	21 dB
Other inband carrier	9 or 10	41 dB

22. 625k-MC base station radio transmission and reception

The BS shall follow all procedures as specified in Clause 6 of ATIS-0700004.2005 with supplemental definitions listed herein to support modulation formats 24-QAM, 32-QAM, and 64-QAM and ModClasses 9 and 10. The terminology used for BS radio compliance specifications is defined in 5.1.1 of ATIS-0700004.2005.

22.1 625k-MC base station transmitter modulation accuracy

The modulation accuracy of the 625k-MC BS transmitter shall be in accordance with the Specifications in 6.3.2.6 of ATIS-0700004.2005 for all Modulation formats including 24-QAM, 32-QAM, and 64-QAM.

22.2 625k-MC base station receiver reference sensitivity level

Reference sensitivity level of 625k-MC BS radio shall be as defined in Table 450 for ModClass 9 and 10 and shall be in Table 50 of ATIS-0700004.2005 for all other ModClasses 0 to 7.

Table 450—BS Reference sensitivity for FER = 10⁻² (Uplink)

Mod Class	Nominal reference sensitivity (dBm)
8	−93.8
9	−91.8
10	−87.2

22.3 625k-MC BS receiver SINR estimation accuracy

Acceptable SINR estimation accuracy Report Values for the 625k-MC BS receiver shall be as in Table 53 of ATIS-0700004.2005 for all ModClasses except ModClass 8, 9, and 10, whose acceptable SINR value shall be as defined in Table 451.

Table 451—Acceptable SINR report values for ModClass 8, 9, and 10 at BS

Input SINR (dB)	5 th percentile (dB)	95 th percentile (dB)
$S < -5$ dB	No minimum	−2 dB
$-5 \leq S < 29$ dB	$S - 3$ dB	$S + 3$ dB
$S \geq 29$ dB	26 dB	$S + 3$ dB

23. 625k-MC L2 MAC Protocol Sublayer Specification

The 625k-MC L2 MAC protocol shall be as specified in Clause 7 of ATIS-0700004.2005 *with the supplemental changes outlined herein*.

To support broadcast services, UT and BS L2 MAC protocol defines additional message: *Short Message Broadcast* transported by logical channel CCH.

23.1 Logical channels

The relationship between logical channels, messages, and burst types shall be as described in Table 55 of ATIS-0700004.2005. The short message broadcast message shall be a standard downlink burst as described in Table 452.

Table 452—Logical channels, messages, and burst types

Logical channel	Message	Burst Types
BCH	Frequency synch message	frequency synchronization
	timing synch message	timing synchronization
	broadcast message	broadcast
PCH	page message	page
RACH	request access message	standard uplink
	access assignment message	standard downlink
CCH	configuration request message	configuration request
	configuration message	standard downlink
	short message broadcast	standard downlink
TCH	uplink traffic	standard uplink
	downlink traffic	standard downlink
FACCH	current modulation class (odd RFN's)	standard uplink
	additional available power (even RFN's)	standard uplink
	recommended modulation class (odd RFN's)	standard downlink
	current modulation class (even RFN's)	standard downlink

23.1.1 Short Message Broadcast (SMB)

This complete subclause is added before the end of subclause 7.3.4 of ATIS-0700004.

Short Message Broadcast (SMB) shall conform to Table 453.

Table 453—SMB fields

Field Name	Number of bits
Minimum protocol	4
AFN	10
Network Operator ID	8
Data flag	4
Sequence number	7
Length	13
Null flag	1
Reserved	10
Data	48
Total	105

- Minimum protocol: Minimum protocol shall be the same as that for CM in 7.3.4. 2 of ATIS-0700004.2005.
- AFN: The 10 least significant bits of the Absolute Frame Number on which short message broadcast channel was transmitted.
- Network Operator ID: Network Operator ID shall be the same as that for CM in 7.3.4. 2 of ATIS-0700004.2005.
- Data flag: Data flag shall identify the broadcast data among the 16 different sets available
- Sequence number: Sequence number shall specify the sequence number of the broadcast packet where the first packet in the sequence shall be 0.
- Length: Shall be the total length of the broadcast message in bits.
- Null flag: Shows that BS does not have any data to transmit.
- Data: Broadcast Data for transmission.

23.1.2 Fast Associated Control Channel (FACCH)

The Fast Associated Control Channel (FACCH) shall as defined in subclause 7.3.6 of ATIS-0700004.2005. The mapping from FACCH on TCH burst to ModClass 0 to ModClass 8 shall be as defined in Table 81 of ATIS-0700004.2005. The mapping from FACCH to ModClass 9 and ModClass 10 shall be as specified in Table 454.

Table 454—FACCH to ModClasses 9 and 10 Mapping

FACCH Message (b3,b2,b1,b0)	ModClass
1001	9
1010	10
1011–1111	Reserved

23.2 625k-MC minimized RMU header

23.2.1 Message format for AA-cts and AA-short

The RMU header defined in ATIS-0700004.2005 shall be referred to as the *basic RMU header*. For small burst sizes, the *basic RMU header presents significant overhead*. For small burst sizes, UT and BS protocols of the 625k-MC mode shall support *minimized RMU header*. The minimized RMU header removes the ARQ acknowledgement field from bursts in which the successful receipt of the reverse-link burst by the peer can implicitly signal correct reception of forward-link data. In addition, the minimized RMU header incorporates only the 8 LSBs of the ARQ sequence number when it is possible to infer the full 13-bit sequence number from the 8 LSBs on the peer. A new ARQ format field is added to the minimized RMU header to indicate whether the header carries an explicit or implicit acknowledgement, and whether the full sequence number or only the 8 LSBs is present. The following subclauses outline the supplemental changes to the baseline L2 MAC Protocol sublayer specifications of ATIS-0700004.2005 when either UT or BS use *minimized RMU header*.

The message format for AA-cts and AA-short shall conform to Table 68 of ATIS-0700004.2005, as augmented in Table 455.

Table 455—Message format for subtypes AA-cts and AA-short

Field	# of bits	Bit positions
(AA common fields)	30	0:29
FrameDec	4	30:33
TchIndex	3	34:36
ModClassUp	4	37:40
modClassDown	4	41:44
Resource	6	45:50
<u>rmuHeaderType</u>	<u>1</u>	<u>51</u>
Reserved	<u>53</u>	<u>52:104</u>
Total	105	

AA common fields consist of the first 5 fields on Table 65 of ATIS-0700004.2005.

The definitions of frameDec, tchIndex, modClassUp, modClassDown, and resource shall be as defined in 7.3.5.1.2.3.1 of ATIS-0700004.2005. The rmuHeaderType field is defined below.

- rmuHeaderType: Shall indicate whether the basic RMU header or minimized RMU header shall be used for the TCH bursts of the stream opened by the AA message. If rmuHeaderType = 0, the basic RMU header shall be used. Else, the minimized RMU header shall be used. The BS shall set the RMU header type field to 0 unless the BS has selected in the Reg Params registration message use of a protocol version that defines the minimized RMU header.

23.2.2 625k-MC RMU field definition

Shall be as specified in 7.3.7.2.1 of ATIS-0700004.2005 with the augmentation shown below.

The structure of AM RMU shall conform to Figure 30 of ATIS-0700004.2005. The gray portion is the RMU Header. The basic RMU header shall be as shown in Figure 30 of ATIS-0700004.2005. The minimized RMU header may be used instead, at the discretion of the BS, if both the BS and UT support it. The header shall be present in every AM RMU. The white portion is the RMU Payload.

Note that the payload size can change when octets are retransmitted. For example, suppose that the AM octets of RMU 1 are lost, and suppose further that RMU 2 is to retransmit as many of these octets as possible. Since the UM portions of RMU 1 and RMU 2 can be different sizes, it is possible that the set of AM octets in RMU 1 will not fit into RMU 2.

23.2.3 625k-MC header field insertion

Shall be as specified in 7.3.7.2.2 of ATIS-0700004.2005. The RMU header fields shall conform to either the basic RMU header that shall be as specified in Table 83 of ATIS-0700004.2005 or to the minimized RMU header that shall be as specified in Table 457.

Table 456—Basic RMU header fields (see Table 83 of ATIS-0700004.2005)

Field	# of bits	Bit positions	Interpretation
pwrCtrl	1	0	Power Control (downlink) / SINR (uplink) bit
Type	2	1:2	00: payload is pure AM data 01: payload is mixture of AM and UM data 10, 11: Reserved
Seqnum	13	3:15	Sequence number of the first AM octet in the payload. If the AM payload size is 0 octets, this value is ignored.
Ack	12	16:27	RLC acknowledgment

Table 457—Minimized RMU header fields

<u>Field</u>	<u># of bits</u>	<u>Bit positions</u>	<u>Interpretation</u>
<u>pwrCtrl</u>	<u>1</u>	<u>0</u>	<u>Power Control (downlink) / SINR (uplink) bit</u>
<u>Type</u>	<u>1</u>	<u>1</u>	<u>0: payload is pure AM data</u> <u>1: payload is mixture of AM and UM data</u>
<u>arqFormat</u>	<u>1 or 3</u>	<u>2:</u>	<u>Coded indication of the format of the ARO seq num and ack fields.</u> <u>0: 8 bit sequence number, ack field is not present</u> <u>001: 8 bit sequence number, ack field is present</u> <u>111: 13 bit seq num, ack field is not present</u> <u>011: 13 bit seq num, ack field is present</u> <u>101: Reserved</u>
<u>seqNum</u>	<u>8 or 13</u>		<u>Sequence number of the first AM octet in the payload. If the AM payload size is 0 octets, this value is ignored. If the arqFormat field is 0 or 001, this field contains the 8 LSBs of the full sequence number.</u>
<u>Ack</u>	<u>0 or 12</u>		<u>RLC acknowledgment. This field is only present when the arqFormat is 001 or 011.</u>
<u>Total</u>	<u>11, 18, 25, or 30</u>		

23.2.4 UM message insertion

Shall be as specified in 7.3.7.2.3 of ATIS-0700004.2005 with the augmentations shown below.

The L2 MAC has two priority levels for AM data (see Clause 8 of ATIS-0700004.2005). The L2 MAC may prioritize UM messages above the high priority AM data, or below the high priority AM data and above the low priority AM data, or below the low priority AM data.

These respective priority levels for UM messages will be referred to as high, medium and low.

The method that L2 MAC uses to do this prioritization shall be as follows:

- Let N_{tot} be the total size of the RMU in bits (which is a function of modulation class) and N_{hdr} the size of the RLC header in bits (see Figure 30 of ATIS-0700004.2005). If the basic RMU header is used, then $N_{\text{hdr}} = 28$ (see Table 456). If the minimized RMU header is used, then L2 MAC shall ask L2 RLC on each frame whether the RMU header will be 11, 18, 25, or 30 bits (see Table 457). The maximum number of bytes available for UM and AM messages is $N_{\text{rmu}} = \lfloor (N_{\text{tot}} - N_{\text{hdr}}) / 8 \rfloor$ bytes where the floor function $\lfloor x \rfloor$ equals the largest integer less than or equal to x .

The remaining steps and text shall be as specified in 7.3.7.2.3 of ATIS-0700004.2005.

24. 625k-MC L2 RLC Protocol Sublayer Specification

625k-MC L2 RLC protocol shall be as specified in Clause 8 of ATIS-0700004.2005 with the supplemental changes outlined herein.

The RMU header defined in ATIS-0700004.2005 is referred to as the *basic RMU header*. UT and BS of the 625k-MC mode shall support header compression techniques to minimize the *basic RMU header*. The following subclauses outline the supplemental changes to the baseline L2 RLC Protocol sublayer specifications of ATIS-0700004.2005 when either UT or BS use the *minimized RMU header*.

24.1 625k-MC AM RMU

As specified in 8.2.1.1 of ATIS-0700004.2005 *with the additional text as underlined below*:

The AM RMU shall send L3 SDUs and some control messages generated by L2 MAC using the L2 RLC entity described in 8.1.3 L2 RLC Model of ATIS-0700004.2005. Figure 30 of ATIS-0700004.2005 shows the format of the RMU Pay load.

24.1.1 RMU header

The RMU header fields shall be as defined in Table 456 and Table 457.

For the basic RMU header and for the minimized RMU header, which include an explicit acknowledgement, the ack field shall contain the 12 LSBs of a sequence number.

24.2 625k-MC transmit procedure

As specified in 8.5.1 of ATIS-0700004.2005 *with the additional text as underlined below*:

During each frame, L2 MAC shall compute the maximum number of octets n_{max} available for transmitting high priority AM octets (see 8.1.3.3 Transmit ARQ of ATIS-0700004.2005), by considering the size of the transmit burst's RMU payload, how many bits are needed for the RMU header, and how many octets are needed for high priority UM messages. The number of bits needed for the RMU header is constant (28) if the basic RMU header is used, and variable if the minimized RMU header is used. Similarly, L2 MAC shall compute $n_{reserve}$, the number of octets desired for medium priority UM messages if L2 RLC has less than n_{max} high priority AM octets. If the number of high priority AM octets is less than $n_{max} - n_{reserve}$, L2 RLC may use the remaining space to transmit any low priority AM octets contiguous to the sequence of high priority AM octets. The Transmit ARQ shall also set seqNum and ack in the RMU header. If the basic RMU header is used, seqNum shall be set to the sequence number of the first octet written to the payload, ack shall be set to the S – 1 LSBs of the value of pAck. If the minimized RMU header is used, L2 RLC shall determine whether to put the full sequence number or only the LSBs into the RMU header, and whether to include or exclude the ack. The L2 RLC shall set argFormat to indicate whether the RMU header contains the full sequence number or the LSBs, and whether the ack is present or implicit (see Table 457).

If the minimized RMU header is used, the RMU header shall have only the 8 LSBs of the sequence number if $pTransmit -_m pAckPeer < 128$ and $pSweep = pAckPeer$. Otherwise the full sequence number shall be used.

If the minimized RMU header is used, then the RMU header shall not include the ack if the following three conditions are met:

- 1) $pAck =_m pReceive$
- 2) the last received RMU was decoded successfully

- 3) the AM data length of the last received RMU payload is zero, or the sequence number of the last byte of the last received RMU payload = $pAck -_m 1$

Otherwise the ack shall be included in the RMU header.

The ack from the peer receive entity shall be processed and the transmit state variable shall be updated as described in 8.5.2.1 of ATIS-0700004.2005 prior to the execution of the transmit states in 8.5.1.1 of ATIS-0700004.2005.

24.2.1 1.1.1625k-MC transmit state execution

Transmit state execution shall be as specified in 8.5.1.1 *with the additional text as underlined in the last bullet point:*

The execution of the Transmit ARQ shall follow the following steps:

- **If $nRetransmit > 0$ (Retransmit state)**

- 1) Set $n := \min(nRetransmit, n_{max})$.
- 2) Send the segment of octets in

$$[pRetransmit, pRetransmit +_m (n-1)]_m$$

- 3) Update $mapTx$ as described in 8.5.1.2 mapTx Update Procedure of ATIS-0700004.2005.
- 4) Set $pRetransmit :=_m pRetransmit + n$.
- 5) Set $nRetransmit := nRetransmit - n$.

- **Else if $nTransmit > 0$ and $pTransmit -_m pAckPeer < 2^{S-1} - 1$ (Transmit state).**

- 1) Set $n := \min(nTransmit, n_{max} - n_{reserve}, 2^{S-1} - (pTransmit -_m pAckPeer) - 1)$.
- 2) Send the segment of octets in

$$[pTransmit, pTransmit +_m (n-1)]_m$$

- 3) Update $mapTx$ as described in 8.5.1.2 mapTx Update Procedure of ATIS-0700004.2005.
- 4) Set $pTransmit :=_m pTransmit + n$.
- 5) Set $nTransmit := nTransmit - n$.
- 6) Cancel timer $T_{shutdown}$ if it is running.

- **Else if $pAckPeer \neq pTransmit$ (Sweep state)**

- 1) If $pSweep <_m pAckPeer$, set $pSweep := pAckPeer$.
- 2) If $pSweep \geq_ m pTransmit$, set $pSweep := pAckPeer$.
- 3) Set $n := \min(pTransmit -_m pSweep, n_{max} - n_{reserve})$.
- 4) Send the segment of octets in $[pSweep, pSweep +_m (n-1)]_m$.
- 5) If the basic RMU is used, update $mapTx$ as described in 8.5.1.2 mapTx Update Procedure of ATIS-0700004.2005. Else, when the minimized RMU header is used, if $RFN - peer\ TCH\ decode - local\ TCH\ decode > mapTx(pSweep)$, update $mapTx$ as described in 8.5.1.2 mapTx Update Procedure of ATIS-0700004.2005.
- 6) Set $pSweep :=_m pSweep + n$.

- Else nothing to transmit (Idle State)

24.2.2 mapTx Update procedure

As defined in 8.5.1.2 of ATIS-0700004.2005.

24.3 Receive procedure

As specified in 8.5.2 of ATIS-0700004.2005 *with the additional text as underlined below*:

24.3.1 Receive task execution

During each frame, the RMU Demux shall deliver to the Receive ARQ the AM payload (possibly 0 octets) as well as the RMU header fields seqNum and ack unless there is a CRC error. (In case of a CRC error, none of them shall be delivered to the AM unit, and the following tasks are not executed.)

If the minimized RMU header is used, then the receiver shall check the arqFormat RMU header field to determine whether the sequence number and ack fields need to be expanded to full length. If arqFormat indicates that the RMU header contains only the 8 LSBs of the sequence number, then the receiver shall recreate the 13 bit sequence number with the following formulas. In these formulas, the & operator indicates a bitwise AND operation, and the ~ operator indicates bitwise negation.

if $\text{seqNum}_{8\text{bits}} \geq ((p\text{Ack} - {}_m 128) \& 0\text{xff})$

$\text{seqNum}_{13\text{bits}} = ((p\text{Ack} - {}_m 128) \& \sim 0\text{xff}) + \text{seqNum}_{8\text{bits}}$

else

$\text{seqNum}_{13\text{bits}} = ((p\text{Ack} - {}_m 128) \& \sim 0\text{xff}) + 0\text{x100} + \text{seqNum}_{8\text{bits}}$

If arqFormat indicates that the RMU header has no ack, then the receiver shall recreate the ack as the following steps:

- 1) Set $\text{AckRFN} = \text{RFN} - N_{\text{turnaround}}$
Here, RFN shows the RFN in which the burst was received.
- 2) Find the range of mapTx elements = AckRFN
- 3) Set ack to the 12 LSBs of the sequence number of the next mapTx element beyond the range identified in step 2).

The receiver shall perform the following three tasks.

Task 1: receive the data in the AM payload

- 1) Set $\text{first} := \text{seqNum}$.
- 2) Set $\text{last} := {}_m \text{seqNum} + \text{size of payload}$.
- 3) If $\text{last} \geq {}_m p\text{Ack} + 2^{S-1}$ and $\text{first} \geq {}_m p\text{Ack}$.
 - i) Recommend forced shut down ($E_{\text{seqNum\_OB}}$ error).
 - ii) Go to Task 2.
- 4) If $\text{last} \leq {}_m p\text{Ack}$, ignore the AM octets in this RMU and go to Task 2.
- 5) Store data in appropriate location.
- 6) Cancel timer T_{shutdown} if it is running.
- 7) If $\text{last} > {}_m p\text{Receive}$, set $p\text{Receive} := {}_m \text{last}$.
- 8) If $\text{first} \leq {}_m p\text{Ack}$,
 - i Find p ($0 \leq p < 2^S$), the first sequence number that satisfies $p \geq {}_m \text{last}$ and $\text{mapRx}[p] = 0$ (that is, there is not a $q < {}_m p$ such that $q \geq {}_m \text{last}$ and $\text{mapRx}[q] = 0$).
 - ii Set $\text{mapRx}[k] = 0$ for $k \in [p\text{Ack}, p - {}_m 1]_m$.
 - iii Deliver and record octets in $[p\text{Ack}, p - {}_m 1]_m$ to L3.

- iv) Set $pAck := p$.
 - v) Cancel timer $T_{receive}$.
 - vi) If $pAck \neq pReceive$, restart timer $T_{receive}$.
- Else (that is, $first >_m pAck$) Set $mapRx[k] = 1$ for $k \in [first, last -_m 1]_m$.

Task 2: Process ack in RMU header (for transmit side)

- 1) ack is $S - 1$ bits only. Convert it to S bits:

If $ack <_m pAckPeer$, set $ack :=_m ack + 2^{S-1}$.

- 2) If $ack >_m pTransmit$
 - i) Recommend forced shut down (error E_{ack_OB}).
 - ii) Go to Task 3.
- 3) If $ack >_m pAckPeer$
 - i) Set $mapTx[k] = -1$ for $k \in [pAckPeer, ack -_m 1]_m$.
 - ii) If $pSweep <_m ack$ Set $pSweep :=_m ack$.
 - iii) Set $pAckPeer :=_m ack$.
 - iv) Cancel timer $T_{transmit}$.
 - v) If $pAckPeer \neq pTransmit$, restart timer $T_{transmit}$.
 - vi) If $nRetransmit > 0$ and $pRetransmit <_m ack$.

Perform a mapTx retransmit update as described in 8.5.2.2 *mapTx Retransmit Update* of ATIS-0700004.2005.

Task 3: check if stream is ready to end

If timer $T_{shutDown}$ is not running

If $nTransmit = 0$, $pAckPeer = pTransmit$, $pAck = pReceive$, a complete packet has been transmitted, a complete packet has been received, and pAck has not changed for at least 1 frame, start timer $T_{shutDown}$

24.3.2 MapTx retransmit update

As defined in 8.5.2.2 of ATIS-0700004.2005.

24.3.3 Reset procedure

As defined in Sect. 8.5.2.3 of ATIS-0700004.2005.

25. 625k-MC L3 Protocol Specification

The L3 protocol shall be as specified in Clause 9 of ATIS-0700004.2005. The Protocol Version Field Setting shall be as defined in Table 98 of ATIS-0700004.2005, with the addition that 625k-MC shall use protocol version field setting value 6.

26. 625k-MC Protocol layer primitives (informative)

The protocol layer primitive shall be as specified in Clause 10 of ATIS-0700004.2005. To support broadcast and multicast services, following primitives are added to individual interfaces in Clause 10 of ATIS-0700004.2005.

26.1 Interface list

The interface list shall be as specified in 10.3.1 of ATIS-0700004.2005.

26.2 Individual interfaces

The basic individual interface primitives shall be as specified in 10.3.2 of ATIS-0700004.2005 along with additional primitives as defined in 26.2.1 to 26.2.6.

26.2.1 L2 MAC to L3 RM Interface Primitives

The L2 MAC to L3 RM Interface Primitives shall be as specified in 10.3.2.10 of ATIS-0700004.2005 *with the following additional primitive described herein.*

Primitive list:

— L2MacUtBCData.ind

Table 458—L2MacUtBCData.ind

<u>L2MacUtBCData.ind</u>	
<u>Applies to:</u>	<u>UT only</u>
<u>Associated primitives:</u>	<u>L2MacBsBCData.req (on BS)</u>
<u>Description:</u> indication from UT L2 MAC to UT L3 RM that incoming broadcast data packets are available	
<u>Parameters:</u> • <u>L2MacUtBcDataInfoSort -- the information of the broadcast data (struct)</u> • <u>L2MacUtBcDataInfoSort :: MinimumProtocol -- a lowest version of the 625k-MC protocol that the BS can support</u> • <u>L2MacUtBcDataInfoSort :: NetworkOperationID -- a identification of the network operator that owns the BS</u> • <u>L2MacUtBcDataInfoSort :: DataFlag -- a flag categories of broadcast data</u> • <u>L2MacUtBcDataInfoSort :: SequenceNumber -- a sequence number of broadcast data</u> • <u>L2MacUtBcDataInfoSort :: Length -- a length of broadcast delivery data</u>	

26.2.2 L3 RM to L2 MAC interface primitives

The L3 RM to L2 MAC Interface Primitives shall be as specified in 10.3.2.11 of ATIS-0700004.2005 *with the following additional primitive described herein.*

Primitive list:

— L2MacBsBCData.req

Table 459—L2MacBsBCData.req

<u>L2MacBsBCData.req</u>	
<u>Applies to:</u>	<u>BS Only</u>
<u>Associated primitives:</u>	<u>L2MacUtBCData.ind (on UT) (on peer)</u>
<u>Description:</u> <u>request from BS L3 RM to BS L2 MAC to deliver broadcast data packets across the air interface to UT L3 RM</u>	
<u>Parameters:</u> • <u>L2MacBsBcDataInfoSort -- the information of the broadcast data (struct)</u> • <u>L2MacBsBcDataInfoSort :: MinimumProtocol -- a lowest version of the 625k-MC protocol that the BS can support</u> • <u>L2MacBsBcDataInfoSort :: NetworkOperationID -- a identification of the network operator that owns the BS</u> • <u>L2MacBsBcDataInfoSort :: DataFlag -- a flag categories of broadcast data</u> • <u>L2MacBsBcDataInfoSort :: SequenceNumber -- a sequence number of broadcast data</u> • <u>L2MacBsBcDataInfoSort :: Length -- a length of broadcast delivery data</u>	

26.2.3 L3 RM to L3 CM interface primitives

The L3 RM to L3 CM Interface Primitives shall be as specified in 10.3.2.18 of ATIS-0700004.2005 *with the following additional primitives described herein.*

Primitive list:

- L3RmBsMCStart.ind
- L3RmUtMCData.ind
- L3RmUtBCData.ind

Table 460—L3RmBsMCStart.ind

<u>L3RmBsMCStart.ind</u>	
<u>Applies to:</u>	<u>BS only</u>
<u>Associated primitives:</u>	<u>L3RmBsMcStart.resp/L3RmUtMCStart.req (on peer)</u>
<u>Description:</u> <u>indication from BS L3 RM to BS L3 CM that a new multicast service has been allowed</u>	
<u>Parameters:</u> • <u>MCstrmID - desired multicast information is requested across the L4.</u>	

Table 461—L3RmUtMCData.ind

<u>L3RmUtMCData.ind</u>	
<u>Applies to:</u>	<u>UT only</u>

Table 461—L3RmUtMCData.ind (continued)

<u>L3RmUtMCData.ind</u>	
<u>Associated primitives:</u>	<u>L3RmBsMCData.req (on BS) (on peer)</u>
<u>Description:</u> indication from UT L3 RM to UT L3 CM that incoming delivered multicast packets are available	
<u>Parameters:</u> • <u>PktQueue</u> -- a list of multicast data packets received across the air interface from the peer BS L3 CM entity	

Table 462—L3RmUtBCData.ind

<u>L3RmUtBCData.ind</u>	
<u>Applies to:</u>	<u>UT only</u>
<u>Associated primitives:</u>	<u>L3RmBsBCData.req (on BS) (on peer)</u>
<u>Description:</u> Indication from UT L3 RM to UT L3 CM that incoming broadcast data packets are available	
<u>Parameters:</u> • <u>PktQueue</u> -- a list of broadcast data packets received across the air interface from the peer UT L3 RM entity • <u>MinimumProtocol</u> -- a lowest version of the 625k-MC protocol that the BS can support • <u>NetworkOperationID</u> -- a identification of the network operator that owns the BS • <u>DataFlag</u> -- a flag categories of broadcast data • <u>Length</u> -- a length of broadcast delivery data	

26.2.4 L3 CM to L3 RM interface primitives

The L3 CM to L3 RM Interface Primitives shall be as specified in 10.3.2.19 of ATIS-0700004.2005 *with the following additional primitives described herein.*

Primitive list:

- == L3RMUtMCStart.req
- == L3RmBsMCStart.resp
- == L3RmBsMCData.req
- == L3RmBsBCData.req

Table 463—L3RmUtMCStart.req

<u>L3RmUtMCStart.req</u>	
<u>Applies to:</u>	<u>UT only</u>
<u>Associated primitives:</u>	<u>L3RmBsMCStart.ind (on BS) (on peer)</u>

Table 463—L3RmUtMCStart.req (continued)

<u>L3RmUtMCStart.req</u>	
<u>Description:</u> <u>request from UT L3 CM to UT L3 RM for a multicast service to allow communication across the air interface</u>	
<u>Parameters:</u> : <u>MCstrmID - desired multicast information is tunneled across the air interface</u>	

Table 464—L3RmBsMCStart.resp

<u>L3RmBsMCStart.resp</u>	
<u>Applies to:</u>	<u>BS only</u>
<u>Associated primitives:</u>	<u>L3RmBsMCStart.ind</u>
<u>Description:</u> <u>response from BS L3 CM to BS L3 RM allowance whether BS L3 CM was able to accept a multicast service in response to the L3RmStart.ind from BS L3 RM</u>	
<u>Parameters:</u> : <u>Boolean -- was the connection started</u>	

Table 465—L3RmBsMCData.req

<u>L3RmBsMCData.req</u>	
<u>Applies to:</u>	<u>BS only</u>
<u>Associated primitives:</u>	<u>L3RmUtMCData.ind (on UT) (on peer)</u>
<u>Description:</u> <u>request to UT L3 RM to deliver multicast data packets across the air interface to BS L3 CM</u>	
<u>Parameters:</u> : <u>PktQueue -- a list of multicast data packets to deliver across the air interface to the BS L3 CM entity</u>	

Table 466—L3RmBsBCData.req

<u>L3RmBsBCData.req</u>	
<u>Applies to:</u>	<u>BS only</u>
<u>Associated primitives:</u>	<u>L3RmUtBCData.ind (on UT) (on peer)</u>

Table 466—L3RmBsBCData.req (continued)

<u>L3RmBsBCData.req</u>	
Description: request from BS L3 CM to BS L3 RM to deliver broadcast data packets across the air interface to UT L3 CM	
Parameters:	
•	<u>PktQueue</u> -- a list of broadcast data packets to send across the air interface to the UT L3 CM entity
•	<u>MinimumProtocol</u> -- a lowest version of the 625k-MC protocol that the BS can support
•	<u>NetworkOperationID</u> -- a identification of the network operator that owns the BS
•	<u>DataFlag</u> -- a flag categories of broadcast data o <u>SequenceNumber</u> -- a sequence number of broadcast data
•	<u>Length</u> -- a length of broadcast delivery data

26.2.5 L3 CM to L4 interface primitives

The L3 CM to L4 Interface Primitives shall be as specified in 10.3.2.20 of ATIS-0700004.2005 *with the following additional primitives described herein.*

Primitive list:

- == L3CmBsMCStart.ind
- == L3CmUtMCData.ind
- == L3CmUtBCData.ind

Table 467—L3CmBsMCStart.ind

<u>L3CmBsMCStart.ind</u>	
<u>Applies to:</u>	<u>BS only</u>
<u>Associated primitives:</u>	<u>L3CmBsMCStart.resp</u>
Description: indication from BS L3 CM to BS L4 to confirm a multicast service	
Parameters:	
•	<u>MCstrmID</u> - desired multicast information is requested.

Table 468—L3CmUtMCData.ind

<u>L3CmUtMCData.ind</u>	
<u>Applies to:</u>	<u>UT only</u>
<u>Associated primitives:</u>	<u>L3CmBsMCData.req (on BS) (on peer)</u>
Description: indication from UT L3 CM to UT L4 that incoming multicast data packets are available	

Table 468—L3CmUtMCData.ind (continued)

<u>L3CmUtMCData.ind</u>	
<u>Parameters:</u>	
•	<u>PktQueue</u> -- a list of data packets received across the air interface from the peer L4 entity

Table 469—L3CmUtBCData.ind

<u>L3CmUtBCData.ind</u>	
<u>Applies to:</u>	UT only
<u>Associated primitives:</u>	L3CmBsBCData.req (on BS)
<u>Description:</u> indication from UT L3 CM to UT L4 that incoming broadcast data packets are available	
<u>Parameters:</u>	
•	<u>PktQueue</u> -- a list of broadcast data packets received across the air interface from the peer L3 CM entity
•	<u>DataFlag</u> -- a flag categories of broadcast data
•	<u>Length</u> -- a length of broadcast delivery data

26.2.6 L4 to L3 CM interface primitives

The L4 to L3 CM interface primitives shall be as specified in 10.3.2.21 of ATIS-0700004.2005 *with the following additional primitives described herein.*

Primitive list:

- L3CmUtMCStart.req
- L3CmBsMCStart.resp
- L3CmMCData.req
- L3CmBsBCData.req

Table 470—L3CmUtMCStart.req

<u>L3CmUtMCStart.req</u>	
<u>Applies to:</u>	<u>UT only</u>
<u>Associated primitives:</u>	<u>None</u>
<u>Description:</u> <u>request from UT L4 to UT L3 CM to start a multicast streaming across the air interface</u>	
<u>Parameters:</u>	
—	<u>MCstrmID</u> - desired multicast information is tunneled across the air interface

Table 471—L3CmBsMCStart.resp

<u>L3CmBsMCStart.resp</u>	
<u>Applies to:</u>	<u>BS only</u>
<u>Associated primitives:</u>	<u>L3CmBsMCStart.ind</u>
<u>Description:</u> <u>response from BS L4 to BS L3 CM allowance whether BS L4 was able to accept a multicast service in response to the L3CmBsMCStart.ind from BS L3 CM</u>	
<u>Parameters:</u> • <u>Boolean -- was the connection started</u>	

Table 472—L3CmBsMCData.req

<u>L3CmBsMCData.req</u>	
<u>Applies to:</u>	<u>BS only</u>
<u>Associated primitives:</u>	<u>L3CmUtMCData.ind (on peer)</u>
<u>Description:</u> <u>request from BS L4 to BS L3 CM to deliver multicast data packets across the air interface to UT L4</u>	
<u>Parameters:</u> • <u>PktQueue -- a list of multicast data packets to deliver across the air interface to the UT L4 entity</u>	

Table 473—L3CmBsBCData.req

<u>L3CmBsBCData.req</u>	
<u>Applies to:</u>	<u>BS only</u>
<u>Associated primitives:</u>	<u>L3CmUtBCData.ind (on UT) (on peer)</u>
<u>Description:</u> <u>request from BS L4 to BS L3 CM to deliver broadcast data packets across the air interface to UT L4</u>	
<u>Parameters:</u> • <u>PktQueue -- a list of broadcast data packets to deliver across the air interface to the UT L4 entity</u> • <u>DataFlag -- a flag categories of broadcast data</u> • <u>Length -- a length of broadcast delivery data</u>	

27. 625k-MC QoS Enhancements

This clause is in addition to the baseline specification ATIS-0700004.2005.

This clause describes the QoS enhancements of 625k-MC mode.

27.1 Classes of services

The 625k-MC system is optimized to meet the Quality of Service (QoS) needs of applications by controlling the air interface sessions. QoS L3 AM control message is used between BS and UT to handle the changes in session's QoS requirements. Different sessions (connections) within a UT may have different QoS markings same as DSCP (Differential Service Code Point from RFC2474) markings, which are conveyed by QoS message as specified in Table 475. If so, the BS may allot the stream resources available to the UT between its registrations based on the appropriate relative priority. See the description of ibQos (QoS) argument of the L3RmBSDData.req and L3CmBsData.req primitives in subclause 10.3.2.19 and 10.3.2.21 of ATIS-0700004.2005.

To support the Standard DiffServ QoS Models, UT and BS of 625k-MC shall support the three classes of services as defined in Table 474 that incorporate per-hop forwarding behaviors (PHB) in the descending order of priority: Priority 1, Priority 2, and Priority 3. These classes are handled by Expedited Forwarding (EF), Assured Forwarding (AF) and Best Effort (BE) forwarding, respectively. In this table, QoS value refers to 6 MSB (most significant bits) of QoS marking in the QoS message of Table 475.

Table 474—Classes of QoS service

	Class of sessions/users	Class of QoS	QoS value (as listed in Table 475 Qos message)	Example of service
1)	Priority1	Expedited Forwarding (EF) – PHB	101110	VoIP
2)	Priority2	Assured Forwarding (AF) – PHB	001010 / 001100 / 001110 010010 / 010100 / 010110 011010 / 011100 / 011110 100010 / 100100 / 100110	Video Stream, Bandwidth Priority
3)	Priority3	Best Effort (BE) – PHB	000000	Data

The QoS value (or DSCP) for Priority 1 is EF Code Point 101110 (refer IETF RFC 2598), the DSCP for Priority 2 are defined with twelve kinds of AF Code Point (same as defined in IETF RFC 2597), and the DSCP for Priority 3 is default Code Point 000000.

27.2 Session QoS information exchange procedures

BS informs any change in the session's QoS by using "QoS Message." The Formats of QoS Message is shown in 9.5.1 in ATIS-0700004.2005, the sequence is shown in Figure 179, which is inherited from Figure 64 of ATIS-0700004.2005.

"QoS Value" (Table 474) is given by 6 msb of the octet: "QoS Marking" of "QoS Message" in 9.5.1.3.12 of ATIS-0700004.2005. "QoS marking," which is shown in Figure 9.59 of ATIS-0700004.2005, is defined by 8 bits, therefore "QoS Value" (6 bits) is set up as follows.

Table 475—QoS Message

msb		bits						lsb	octet
8	7	6	5	4	3	2	1		
0		0	0	0	1	1	0	0	0
msgID									
0	TLVG length								1
ext bit	(assuming<128bytes)								
0	0	0	0	1	1	0	0	2	
TLVG ID									
QoS marking									3
QoS Value							Reserved		

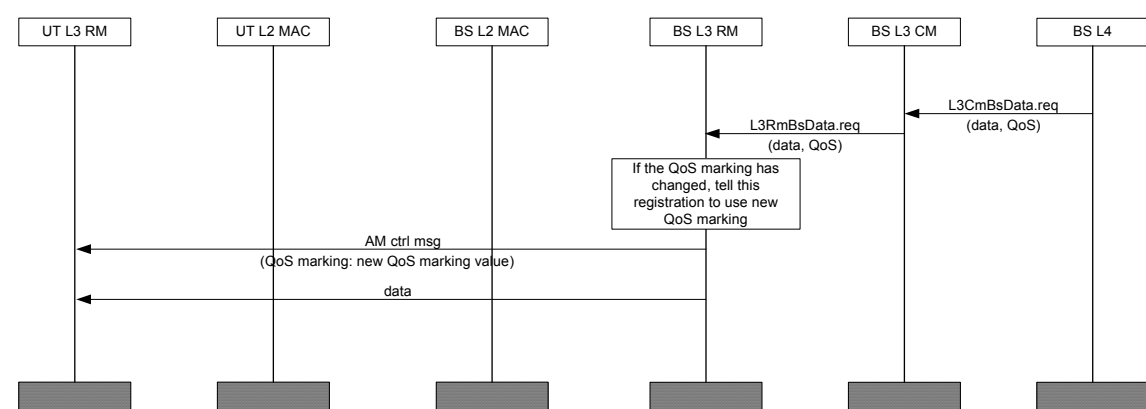


Figure 179—QoS message sequence

When the BS is told by network to change the session's quality of service, BS delivers the QoS L3 AM Control (QoS) Message and thereby notifies a new QoS value to UT.

27.3 QoS priority

As listed in Table 474, there are three Classes of services with different priorities.

For each class, Shutdown Timer (defined in 8.2.2.3 of ATIS-0700004.2005), which can specify the time to hold a stream can be set. Based on the priority, the Shutdown Timer can be set differently for different streams. For high priority data, a large timeout value should be used for the Shutdown Timer. These values are different for different streams.

It can perform differentiation by reservation of the stream according to priority. Generally, longer Shutdown Timer is set up for the sessions with a high class of priority. The values of longer Shutdown Timer may be implementation specific.

27.3.1 QoS Priority 1

Priority 1 of QoS shall be the highest priority of three Classes of Service.

It shall support high quality transmission achieving low packet loss, low latency, and low jitter duration. It is defined as the EF-PHB, which 6 bits from MSB of QoS Marking field in the QoS Message is 101110.

The Shutdown Timer shall be specified to a larger value than that of other classes.

27.3.2 QoS Priority 2

Priority 2 of QoS shall be defined to a priority level lower than Priority 1.

It shall support the higher quality transmission than Priority 3.

It is defined as the AF-PHB, and twelve kinds of QoS Marking field in the QoS Message could be defined and used flexibly in each implementation.

The Shutdown Timer shall be set to a value smaller than Priority 1 and larger than Priority 3.

27.3.3 QoS Priority 3

Priority 3 of QoS shall be defined to Best Effort PHB.

It is defined as the BE-PHB, which 6 bits from MSB of QoS Marking field in the QoS Message is 000000.

The Shutdown Timer shall be specified to a smaller value than that of other classes.

28. 625k-MC Broadcast and Multicast Service (BCMCS) Support Enhancement

This clause is in addition to the baseline specification ATIS-0700004.2005.

28.1 Overview

Broadcast service: Any user can get information from the *BCMCS Contents Server*. Any user can get this information; extra certification is not needed.

Multicast service: Particular user can get various beneficial information from the *BCMCS Content Server*.

28.2 Broadcast service

A BCH superframe shall be as defined in Figure 23 in Chapter 7 of ATIS-0700004.2005. Broadcast service shall be sent using CM burst in the superframe. This CM burst is called *Short Message Broadcast (SMB)*. There is one SMB broadcast in a superframe and the position is always after the B burst. SMB follows the B depending on the BSCC; for example, if BSCC = 3 then SMB is after B3 as shown in Figure 180. SMB channel is only used for short broadcast service. BS shall inform BCMCS contents controller of Broadcast service request. BS gets Broadcast contents data from BCMCS contents server. Broadcast service can send at most 768 bytes. This message has 4 bit data flag to identify the broadcast data among the 16 different sets available. UT receives a part of the data with each superframe, and combines the data after receiving the entire data. Message flow for short message broadcast is shown in Figure 181.

SMB fields are shown in Table 453.

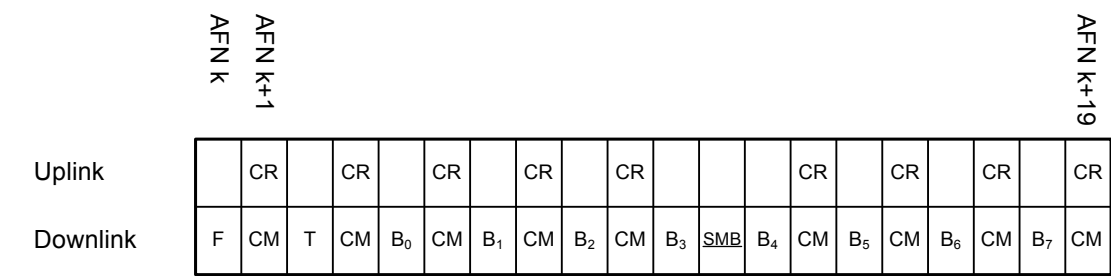


Figure 180—BCH superframe with SMB after B3

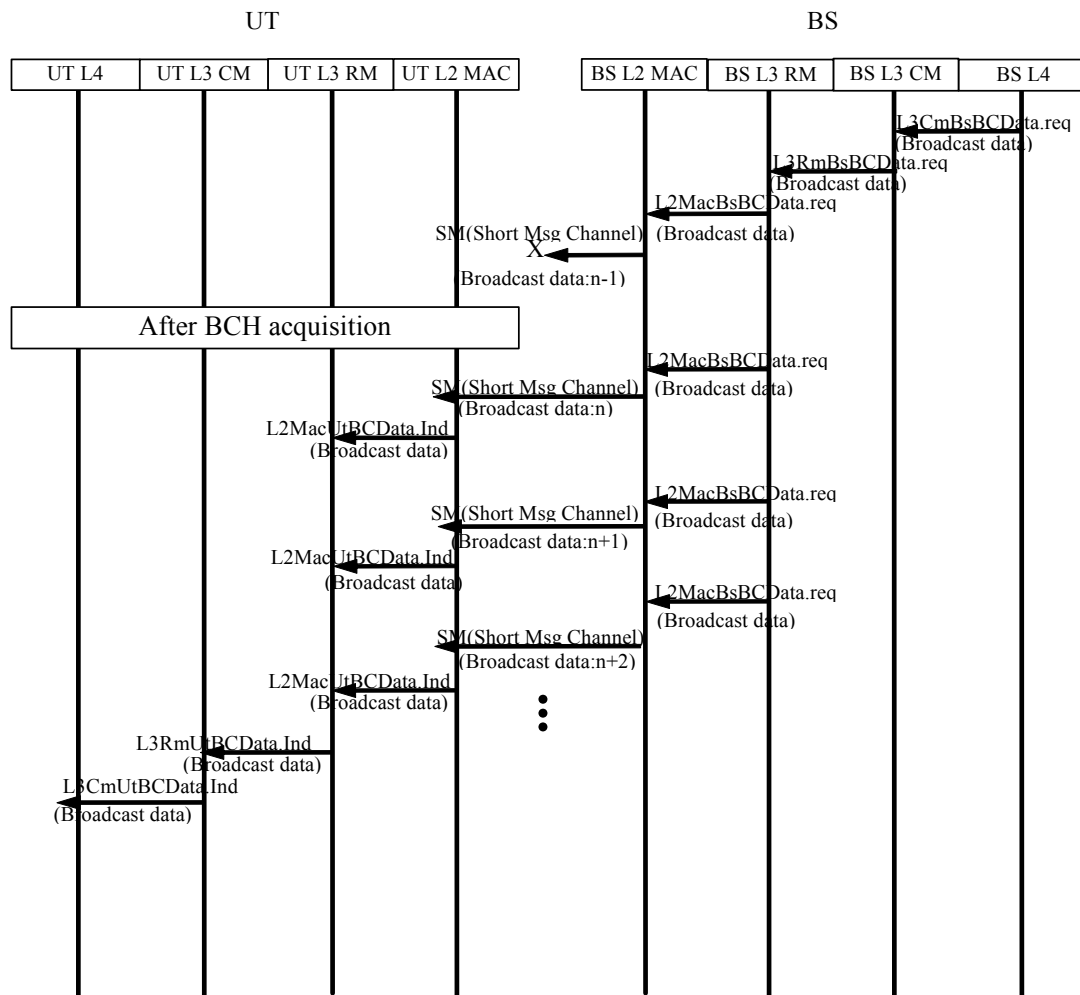


Figure 181—Message Flow for short message channel

28.3 Multicast service

28.3.1 Overview

Multicast channels are made available to many users, therefore the functions of RRC are necessary for individual TCH of multicast channel.

28.3.2 Multicast handshake

To connect to a multicast channel, UT shall send *Multicast stream request* to BS. BS shall send a *Multicast stream response* which includes *resource map* information. Resource map information tells the UT about the TCH channel over which multicast data is transmitted. This TCH channel shall be called the *Multicast channel*. UT changes the receiving TCH channel to Multicast Channel. The messaging handshake for Multicast stream shall conform to Figure 182.

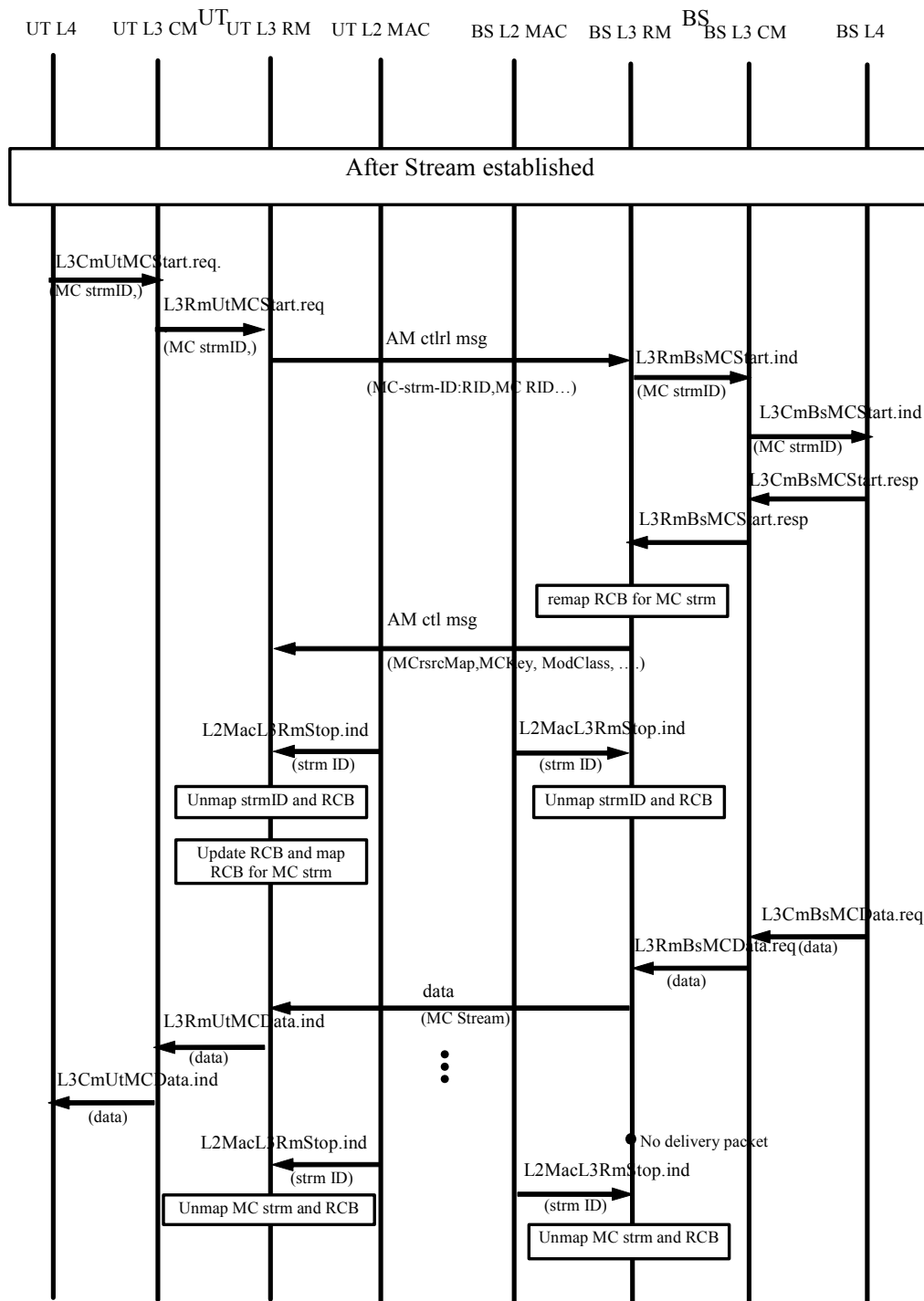


Figure 182—Message Flow for Multicast channel

- To send the MCStream.req to BS, UT opens a stream after TCH connection following the *UT Initiated stream Start* sequence specified in Figure 58 of the ATIS-0700004.2005.
- UT sends information about MCStrmID, RID together with the MCStream.req over the connected TCH. If BS allows the stream to open, it shall respond by sending the MCStream.resp with information necessary for Multicast like MC key, MC ModClass, and MCsrcMap.

- UT and BS shall close the stream following the *Graceful Stream Closure* sequence specified in Figure 66 of ATIS-0700004.2005.
- BS shall transmit Multicast information over the channel specified using the MCsrcMap during MCStream.resp; UT shall receive over the same channel. During Multicast information transmission and reception, RRC functions are not considered. Data shall be transmitted from BS at a constant level with the same ModClass specified during MCStream.resp. The data shall be encrypted at BS and the UT shall decrypt it using the MC key. MC key length shall be chosen between 56 bit and 280 bit for Multicast registration.

In case the UT is unable to receive multicast data or the multicast data finishes, the Multicast session shall be closed following the graceful stream closure sequence as specified in Figure 66 of ATIS-0700004.2005. The BS shall continue to transmit until the data finishes.

28.3.3 Message format

Before the UT receives the multicast data, UT needs to register with the BS using the AM message data, viz., MCStream.req and MCStream.resp. BS uses AM mode also to deliver Multicast data to UT. The RMU header field shall conform to the Table 83 in 7.3.7.2.2 of ATIS-0700004.2005. BS shall use RMU header type as “01”, which indicates “payload is mixture of AM and UM data.” During multicast session, BS and UT shall ignore “power control,” “FACCH,” and “ack” for AM in this BCMCS protocol.

The message format for MCStream.req and MCStream.resp are as shown in Figure 181 and Figure 182, respectively.

28.3.3.1 MCStream.req message

Message ID: 16

TLVG: MCStream.req

TLVG ID: 16

Information Elements:

- IE: Assigned protocol version, 8 bits

Description:

The current protocol version field indicates which revision of the protocol the sending party is currently using. This is used by the UT in the MCStream.req message and by the BS in the MCStream.resp message to inform each other of which protocol version is being used during the multicast stream.

- IE: RID, 15 bits

Description:

The registration ID (RID) is used in various of the RA/AA messages that open streams to identify which registration the stream is for.

- IE: MCstrmID, 17 bits

Description:

The MCstrmID indicates the UT’s desired multicast information which is depended on

delivered contents. When the BS delivers only one multicast channel, MCstrmID value is all zero.

Table 476—MCStream.req format

msb						lsb	
1	2	3	4	5	6	7	8
0	0	0	1	0	0	0	0
msg ID							
0 ext bit	TLVG length (assuming < 128 bytes)						
0	0	0	1	0	0	0	0
msg ID							
Assigned 625K-MC protocol version							
RID							
RID						MCstrmID	
MCstrmID							
MCstrmID							
Reserved							

28.3.3.2 MCStream.resp message

Message ID: 17

TLVG: MCStream.resp

TLVG ID: 17

Information Elements:

- IE: Assigned 625k-MC protocol version, 8 bits

Description:

The assigned protocol version field indicates which revision of the 625k-MC protocol the sending party is requiring the receiving party to use. This is used by the BS in the MCStream.resp message to tell the UT what protocol version to use throughout the multicast registration.

- IE: McsrcMap, 48 bits

Description:

The McsrcMap specifies which conventional channels of the BS the UT is allowed to use for Multicast stream. The LSB of this bit map corresponds to conventional channel ID 0. See A.25 in Annex A of ATIS-0700004.2005 for a Definition of the Conventional Channel ID.

- IE: reject presence, 1 bit

Description:

This reject presence indicates whether the BS chose to reject the multicast registration. When this field is 0, the multicast registration is not rejected. When this field is 1, the multicast registration is rejected by the BS.

- IE: MC reject cause, 5 bit

Description:

This MC reject cause indicates the reason for the multicast rejection. The rejection causes are listed in Table 477 when reject presence field is 1.

Table 477—Multicast connection rejection values

Value	Rejection/Cause
0	Unknown/unspecified
1	The UT's protocol version is too old
2	The UT's protocol version is too new
3	The BS's current traffic load is too high
4	The UT is not registered
5	The BS is not delivered for multicast data
6	The BS is preparing multicast service
7	Invalid MCStreamID value
8–31	Reserved

- IE: MC key size, 5 bits

Description:

The K size field MCKey is used by the BS to specify how many bytes of the (decrypted) multicast security stream secret will be used as the i-SEC encryption secret. If this field is non-zero, the key size is the value of this field plus four, in bytes. If the field is zero, this indicates that for testing purposes multicast security is disabled for the registration.

- IE: MC key, 326-640 bits

Description:

The MC key consists of i-SEC key (K), i-SEC key(Kr), counter value and AFN. This K, AFN and counter value changes every frame. In fact, i-SEC key and AFN are one-on-one relationship.

This K is composed in multiples of 8 bits that total size is between 40 bits and 280 bits. The key size depends on the multicast provider. And AFN size is 26 bits, counter value size is 8 bits, and Kr is 326 bits. Total size is a maximum of 640 bits.

Multicast delivery data is encrypted to use the part of MC key which is shared secret key by the BS. In contrast, the UT identifies the encrypted data to decrypt the same parts of MC key

- IE: MC ModClass, 4 bits

Description:

The MC ModClass indicates multicast delivering ModClass from the BS. This delivery Modclass shall be set by the BS.

Table 478—MCStream.resp format

msb						lsb	
1	2	3	4	5	6	7	8
0	0	0	1	0	0	0	1
msg ID							
0 ext bit	TLVG length (assuming < 128 bytes)						
0	0	0	1	0	0	0	1
msg ID							
Assigned 625K-MC protocol version							
MCsrcMap							
MCsrcMap							
MCsrcMap							
MCsrcMap							
MCsrcMap							
MCsrcMap							
reject presence	MC reject cause					Reserved	
Reserved			MC key size				
MC key							
MC key							
-							
-							
-							
MC ModClass				Reserved			
Reserved							

29. 625k-MC Privacy and Authentication Enhancement

As defined in Clause 11 of ATIS-0700004.2005 *with the additional text as underlined below*:

The symmetric key encryption engine utilized in the AES for AES algorithm shall be as described in FIPS PUB 197.

29.1 Overview

As defined in 11.1 of ATIS-0700004.2005.

29.2 625k-MC Handshake and BS Authentication Protocol, i-HAP

As defined in 11.2 of ATIS-0700004.2005, *with the additional definitions and text for the certificate message fields in subclause 11.2.2 of ATIS-0700004.2005 as underlined below*:

10. *i-SEC Bulk Encryption Algorithm Choices*: Shall be 4 bits. This shall be a bitmap of bulk encryption algorithms supported by the BS. Every supported algorithm shall have a '1' in the relevant bit. For example, if the stream cipher method and the AES modes are supported then this field shall be set to [1,1,0,0].

Bit 0: Stream Cipher (see 11.4.2 of ATIS-0700004.2005).

Bit 1: AES Cipher.

Bit 2: Reserved.

Bit 3: Reserved.

11. *i-SEC Bulk Encryption Parameters*: Shall be 20 bits. This field shall contain 4 subfields, each 5 bits. The k^{th} subfield shall contain parameters related to the k^{th} bulk encryption algorithm presented in Field (10) above. If a bulk encryption algorithm is not supported, then the corresponding subfield shall be set to all zeros. The information in each subfield shall depend on the definition and parameters of the corresponding bulk encryption algorithm.

— For the stream cipher:

- The first subfield shall provide the secret key size in bytes: if this field is all zeros, then the stream cipher is disabled; otherwise, the secret key size is the value represented in first subfield (in little endian format) plus four in bytes.
- The resulting value (in bytes) shall be defined as the maximum value of encryption key size () supported by the BS.
- For example, a BS that supports the stream cipher alone, with a maximum encryption secret key size of 56 bits (7 bytes) has:
 - [1,0,0,0] as its Algorithm Choices field (Field 9 above), and [1,1,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0] as its Bulk Encryption Parameters field. Note that each subfield represents an integer as a bit string encoded with LSB first.

— For the AES cipher:

- The first two bits of the second subfield shall be used to indicate the maximum length of secret key size to be used in the AES cipher. The value [10] indicates 128-bit secret key, [01] indicates 192-bit secret key, and [11] indicates a 256-bit secret key for the cipher. The value [0,0] indicates that the AES cipher is disabled.
- The remaining three bits of the second subfield shall be used to indicate the encryption mode of the AES cipher. If the AES cipher is disabled, then this field shall be set to

- [0,0,0]. The value [1,0,0] indicates that the output feedback (OFB) mode is supported. The remaining three-bit combinations are reserved.
- == For example, a BS that supports the AES cipher alone, with a maximum of 192 bit secret key length and supports the OFB mode has:
- [0,1,0,0] as its Algorithm Choices Field and [0,0,0,0,0;**0,1,1,0,0**; 0,0,0,0,0; 0,0,0,0,0; 0,0,0,0,0]. Note that each subfield represents an integer as a bit string encoded with LSB first.
 - A BS that supports **both** the stream cipher (key size 56 bits) and the AES cipher (key size 256 bits) has:
 - [1,1,0,0] as its Algorithm Choices Field, and [**1,1,0,0,0**;**1,1,1,0,0**; 0,0,0,0,0; 0,0,0,0,0; 0,0,0,0,0] as its Bulk Encryption Parameters field.

With the above additional definitions, the BS certificate message fields are illustrated inFigure 183.

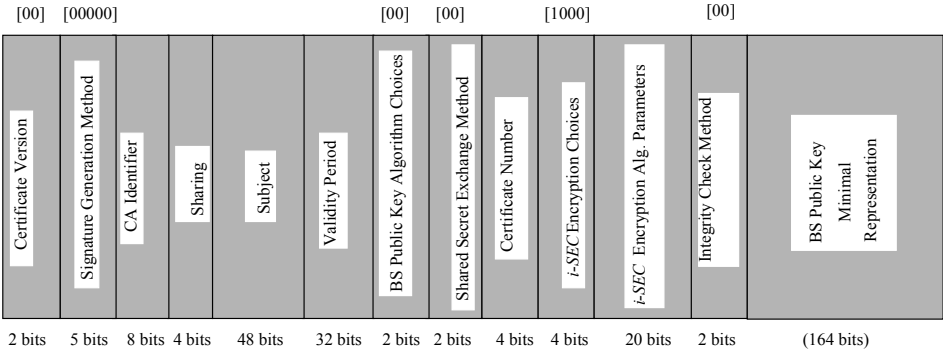


Figure 183—HC-SDMA BS certificate message fields
(Mandatory field values are identified in square brackets. Not to scale.)

29.3 625k-MC Terminal Authentication Protocol, i-TAP

As defined in 11.3 of ATIS-0700004.2005.

29.4 625k-MC Secure Communications Protocol, i-SEC

The Secure Communications Protocol, i-SEC shall be as specified in 11.4 of ATIS-0700004.2005 *with additional support for minimized RMU header and AES as specified herein.*

In ATIS-0700004.2005, the i-SEC protocol is responsible for symmetric encryption of data and control messages based on the shared secret. In 625k-MC mode, BS and UT shall support the AES cipher algorithm in addition to the stream cipher algorithm used in ATIS-0700004.2005.

i-SEC protocol is responsible for the encryption of TCH traffic streams that carry AM data and control messages. The BS certificate provides information about the capabilities of the base station (refer to clause 11.2.2 of ATIS-0700004.2005) in terms of supported algorithms and secret key sizes. In 625k-MC mode, i-SEC shall employ the stream cipher algorithm and the AES algorithm explained in this subclause and shall use a portion⁵² of the shared secret K_r established during i-HAP.

⁵² A disjoint portion of the shared secret shall be used for scrambling in i-TAP

In addition, the BS shall have two thresholds: `regThreshBS` that limits the number of TCH streams that use the same shared secret and `streamThreshBS` that limits the number of bursts per TCH stream. The UT shall have its own thresholds `regThreshUT` and `streamThreshUT` for the same purpose. Refer to clause 11.4.2.2 of ATIS-0700004.2005, Stream Counter Test and clause 11.4.2.6 of ATIS-0700004.2005, Burst Counter Test for details on the use of these thresholds. The UT shall be responsible for closing TCH streams or initiating new registrations based on the updated values of `regThreshUT` and `streamThreshUT` gracefully. The BS shall be responsible for terminating TCH streams and registrations based on its thresholds `streamThreshBS` and `regThreshBS`.

The i-SEC encryption key K shall consist of the L_{key} lowest significant bytes of the shared secret K_r (326 bit long shared secret which shall be exchanged with the current BS public key encryption algorithm). The largest value of the parameter L_{key} supported by a BS shall be presented by the first subfield of field (12) of the BS certificate (refer to Clause 11.2.2 of ATIS-0700004.2005, Base Station Transmits Its Digital Certificate). For the stream cipher, L_{key} shall have a maximum value of 35 bytes, allowing for encryption keys of up to 280 bits. The remaining 46 bits of the shared secret shall be reserved. For the AES cipher, L_{key} shall have a maximum value of 32 bytes, allowing for encryption keys of up to 256 bits. The next 32 bits of the shared secret shall be used for initialization purposes (shared secret bits 0-255 for AES encryption key, and shared secret bits 256-287 are used for initialization). The remaining shared secret bits are reserved. L_{key} can take on only three values: {16,24,32} bytes. The actual length of the i-SEC encryption key is negotiated between the UT and the BS. The remaining 46 bits of the shared secret shall be reserved.

29.4.1 TCH streams

As defined in 11.4.1 of ATIS-0700004.2005.

29.4.2 625k-MC Symmetric Key Stream Cipher Algorithm

As specified in 11.4.2 of ATIS-0700004.2005 *with the following changes to support minimized RMU header.*

29.4.2.1 Initialization Vector Selection

As specified in 11.4.2.1 of ATIS-0700004.2005.

29.4.2.2 Stream Counter Test

As specified in 11.4.2.2 of ATIS-0700004.2005.

29.4.2.3 Determination of Encryption Key from Shared Secret

As specified in 11.4.2.3 of ATIS-0700004.2005.

29.4.2.4 Stream Cipher State initialization

As specified in 11.4.2.4 of ATIS-0700004.2005.

29.4.2.5 Keystream generation

As specified in 11.4.2.5 of ATIS-0700004.2005 *with the additional text as underlined below:*

- The maximum uplink burst is 1000 bits, that consists of 4 pad bits, 28 bits of RMU Header, 119x8 bits of RMU payload and 16 bits of CRC. The CRC shall not be encrypted in i-SEC.
- The maximum downlink burst is 2539 bits, that consists of 7 pad bits, 28 bits of RMU Header, 311x8 bits of RMU payload and 16 bits of CRC. The CRC shall not be encrypted in *i-SEC*.

29.4.2.6 Burst Counter Test

As specified in 11.4.2.6 of ATIS-0700004.2005.

Encryption and Decryption Using i-SEC Keystream

As specified in 11.4.2.7 of ATIS-0700004.2005 with the additional text as underlined below:

i-SEC encryption shall be performed by XORing the RMU header and payload (see 7.3.7.2 RMU MuxFunction and Figure 30 of ATIS-0700004.2005 for definitions) by the *i-SEC* keystream before transmission. Therefore, *i-SEC* encryption is not a bandwidth expanding transformation. The 16 bit CRC shall be calculated over the encrypted bits. The CRC shall not be encrypted. *i-SEC* decryption shall be performed by XORing the received encrypted RMU header and payload octets by the *i-SEC* keystream. The 16 bit CRC check shall be performed on the received encrypted bits before decryption.

In streams using the basic RMU header. The RMU header and payload shall be aligned such that the last payload falls exactly at the end of the relevant keystream buffer. In streams using the minimized RMU header, the last burst bit before the CRC bits shall be aligned with the end of the relevant keystream buffer. During this process, some of the keystream bytes or bits are possibly left unused and they shall be discarded.

For example, consider an uplink burst payload that contains 28 bits of basic RMU header and 3 payload octets {ABC} with a basic RMU header H represented by a bit vector $[h_0, h_1, \dots, h_{27}]$ where h_0 is transmitted first and the header is partitioned into four bytes: $H_0 = [0, 0, 0, 0, h_0, h_1, h_2, h_3]$, $H_1 = [h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}]$, $H_2 = [h_{12}, h_{13}, h_{14}, h_{15}, h_{16}, h_{17}, h_{18}, h_{19}]$, and $H_3 = [h_{20}, h_{21}, h_{22}, h_{23}, h_{24}, h_{25}, h_{26}, h_{27}]$.

29.4.3 625k-MC AES Cipher Algorithm

This complete subclause is added before the end of subclause 11.4 of ATIS-0700004.2005

i-SEC shall employ the symmetric key encryption algorithm for AES. The AES algorithm is defined in FIPS PUB 197. *i-SEC* shall support the AES encryption algorithm. The following subclauses provide the details. Note that the AES cipher can only utilize up to 256 least significant bits of the shared secret (K_r) established during i-HAP as the AES cipher key (K_{AES}).

29.4.3.1 AES Initialization Vector Selection

The *i-SEC* encryption key K_{AES} shall be utilized together with a 128-bit AES initialization vector (IV_{AES}) to initialize the AES cipher operation at the beginning of a stream. IV_{AES} shall be different for each TCH stream over the expected lifetime of the shared secret key.

Let IV denote the 32-bit initialization vector defined in 11.3.3 of the ATIS-0700004.2005 standard for the stream cipher. Let us define the 32-bit vector W:

$$W(n) = K_r(n + 256), \quad n = 0, 1, \dots, 31.$$

Where $W(n)$ denotes the n^{th} bit of W. The 32-bit vector W shall be modified by a component-wise XOR with the IV defined in 11.3.3 of the ATIS-0700004.2005 standard.

$$W(n) = W(n) \text{ XOR } IV(n), \quad n = 0, 1, \dots, 31.$$

The initialization vector for the AES algorithm is defined as the concatenation of W to obtain a length 128-bit vector:

$$IV_{AES} = [W, W, W, W];$$

As a result the true value of IV_{AES} is unknown to parties other than the BS and the UT.

29.4.3.2 AES Stream Counter Test

As specified in 11.4.2.2 of ATIS-0700004.2005.

29.4.3.3 Determination of AES Encryption Key from Shared Secret

The length (in bytes) of *i-SEC* encryption key () shall be defined as L_{key} . The second subfield of Field (12) of the BS certificate shall specify the maximum value of L_{key} supported by the BS. The value of L_{key} shall be in the set {16, 24, 32} bytes for the AES cipher.

The first ($L_{key} \times 8$) least significant bits of the shared secret K_r shall be defined as the encryption key, K_{AES} :

$$K_{AES}(n) = K_r(n), \quad n=0,1,\dots,8L_{key}-1.$$

Where $K_{AES}(n)$ is the n^{th} bit of the *i-SEC* encryption key K_{AES} , and $K_r(n)$ is the n^{th} bit of the *i-SEC* encryption key K_r . For maximally secure deployments, L_{key} may be set to 32. The actual key size used by *i-SEC* shall be negotiated during registration. The BS certificate shall set the maximum length on the key size that the BS supports. The UT shall send the key length to be used for the registration in the UT Params message. Test UTs may use a method to bypass *i-SEC* encryption.

29.4.3.4 AES Cipher State initialization

Definition 1: Let $Y=E(Z;M)$ denote the encryption of a 128 bit message vector M , with an AES key named as Z (128, 192 or 256 bits) as defined in the FIPS Standard 197. The cipher output is a 128 bit vector Y .

Definition 2: Let $M=D(Z;Y)$ denote the decryption of a 128 bit vector Y , with an AES key named as Z (128, 192 or 256 bits) as defined in the FIPS Standard 197.

Each stream shall maintain a single AES encryption engine, and a 128-bit initialization vector (IV_{AES}). The computation initialization vector (IV_{AES}) is defined in 29.4.3.1.

At the beginning of each stream, the AES initialization vector is updated (IV_{AES}) by two successive encryption operations with the AES encryption key (K_{AES}) before encrypting any information bits:

$$IV_{AES} = E(K_{AES}, E(K_{AES}, IV_{AES}));$$

This operation takes place both at the message originator (encryptor) and the message receiver (decryptor).

29.4.3.5 AES keystream generation

After the double encryption operation on IV_{AES} (refer to previous subclause) that precedes the keystream generation procedure, a block of 128 keystream bits $U = [U_0, \dots, U_{127}]$ is generated and the initialization vector IV_{AES} is updated by the following procedure:

$$U = E(K_{AES}, IV_{AES});$$

$$IV_{AES}(n) = U_n \text{ XOR } IV_{AES}(n), \quad n = 0, 1, \dots, 127;$$

Note that each call to the above two-line procedure generates 16 keystream bytes.

On every frame following the RA/AA frame, 448-keystream bytes (28 calls to the above procedure per frame) shall be generated for uplink and downlink encryption. The first 128-keystream bytes (8 calls) shall be allocated for uplink encryption and the remaining 320-keystream bytes (20 calls) are allocated for downlink encryption.

Note that these keystream bytes will be generated regardless of the modulation class employed. The unused keystream bytes shall be disregarded. The encryption/decryption procedure using the keystream is explained in 29.4.3.7.

29.4.3.6 AES Burst Counter Test

The UT shall increment its counter `burstCount` after generating the keystream for a burst. This counter shall be set to zero at every stream start. After incrementing `burstCount`, the UT shall compare the value of the counter to a threshold entitled `streamThreshUT`. The UT application software shall set the value for the stream termination threshold `streamThreshUT`. If `streamThreshUT` is non-zero, and the value of `burstCount` is equal to `streamThreshUT`, then the UT shall close the current stream and shall start a new stream with a new stream secret and IV. If the value of `streamThreshUT` is zero, then the UT shall not force a stream closure.

The BS shall perform the same operations independently by using its own counter `burstCount`, and its own threshold `streamThreshBS`. The `streamThreshBS` and the `streamThreshUT` shall be given a nominal value based on the Class of QoS of the session (see Table 474). When the Class of QoS is EF or AF, a nominal value for `streamThreshBS` and `streamThreshUT` shall be 524 288 bursts. Otherwise, a nominal value for `streamThreshBS` and `streamThreshUT` shall be 20 000 bursts. At a combined uplink/downlink approximate data rate of 700 kilobits/sec (single burst/frame, 200 frames/sec), this nominal value of 20 000 bursts results in a stream closure approximately after 8.75 megabytes of over the air traffic before the initialization vector is changed and the stream cipher is reinitialized. This selection also limits the stream duration to 100 seconds regardless of the data rate.

29.4.3.7 Encryption and Decryption Using i-SEC AES Keystream

As specified in 11.4.2.7 of ATIS-0700004.2005 *with the additional text as underlined below*:

i-SEC encryption shall be performed by XORing the RMU header and payload (see 7.3.7.2 RMU MuxFunction and Figure 30 of ATIS-0700004.2005 for definitions) by the *i-SEC* keystream before transmission. Therefore, *i-SEC* encryption is not a bandwidth expanding transformation. The 16 bit CRC shall be calculated over the encrypted bits. The CRC shall not be encrypted. *i-SEC* decryption shall be performed by XORing the received encrypted RMU header and payload by the *i-SEC* keystream. The 16 bit CRC check shall be performed on the received encrypted bits before decryption.

In streams using the basic RMU header, The RMU header and payload shall be aligned such that the last payload falls exactly at the end of the relevant keystream buffer. In streams using the minimized RMU header, the last burst bit before the CRC bits shall be aligned with the end of the relevant keystream buffer. During this process, some of the keystream bytes or bits are possibly left unused and they shall be discarded.

For example, consider an uplink burst payload that contains 28 bits of basic RMU header and 3 payload octets {ABC} with a basic RMU header H represented by a bit vector $[h_0, h_1, \dots, h_{27}]$ where h_0 is transmitted first and the header is partitioned into four bytes: $H_0 = [0, 0, 0, 0, h_0, h_1, h_2, h_3]$, $H_1 = [h_4, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}]$, $H_2 = [h_{12}, h_{13}, h_{14}, h_{15}, h_{16}, h_{17}, h_{18}, h_{19}]$, and $H_3 = [h_{20}, h_{21}, h_{22}, h_{23}, h_{24}, h_{25}, h_{26}, h_{27}]$.

30. 625k-MC Sleep Mode Control Protocol

This clause is an added to the baseline specification ATIS-0700004.2005.

In Idle mode UT periodically monitors own paging PCH (Refer to 7.3.3 of ATIS-0700004.2005) for wake up requirement. When BS needs to deliver data to UT in Idle Mode, BS shall follow the procedure as specified in 9.3.4.1 of ATIS-0700004.2005. The frequency of pages from BS shall conform to five discrete levels of paging activity levels (Paging States) transitions from high level to low level: Page every frame, Page every 8th frame, Page every 64th frame, Page every 512th frame and No paging at all, as specified in 9.3.4.1.2 of ATIS-0700004.2005. BS and UT shall agree during the registration phase on the length of duration of paging activity before changing to next lower paging activity, thereby allowing the power conservation feature in 625k-MC mode. This feature may be used to plan the hardware resources at UT at powerdown mode.

31. 625k-MC OA & M Radio Network Quality Monitor and Control Enhancement

This clause is an added to the baseline specification ATIS-0700004.2005.

625k-MC network systems provide radio network quality monitoring and control functionality. The MIB of 625k-MC mode comprises of the managed objects, attributes, actions, and notifications required to manage a BS. The definition of these managed objects, attributes, actions, and notifications, as well as their structure, is presented in 31.1.3.

31.1 625k-MC Mode MIB

31.1.1 Overview

This chapter defines a Management Information Base (MIB) module for managing the 625k-MC mode. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). The objects in this MIB are defined using the mechanisms specified in the Structure of Management Information (SMI). The MIB module specified is compliant to SMIV2, which is described in IETF RFC 2578, IETF RFC 2579, and IETF RFC 2580.

31.1.2 625k-MC MIB Structure

The MIB is implemented as a single flat structure.

31.1.3 Definition

```
IEEE802dot20-625k-MC-MIB  DEFINITIONS ::= BEGIN

IMPORTS
    OBJECT-TYPE
        FROM RFC-1212
    enterprises, Gauge, Counter
        FROM RFC1155-SMI
    transmission FROM RFC1213-MIB
    ;

IEEE802dot20-625k-MC-MIB MODULE-IDENTITY
    LAST-UPDATED      ""
    ORGANIZATION      "IEEE 802.20"
    CONTACT-INFO
        "WG URL: ieee802.org/20
        MIB Contact: R. Canchi
        Postal: 2480 N. First St., Suite 280
        San Jose, CA, 95131, USA
        Tel: 408-952-4701
        E-mail:  cradhak@ktrc-na.com"

    DESCRIPTION
        "The MIB private module for IEEE 802.20 entities"

    ::= { iso (1) iso-identified-organization (3) ieee (111) standards-
association-numbered-series-standards (2) lan-man-stds (802) ieee802dot20
(20) ieee802dot20mibs(1) ieee802dot20625kMIB(2) }

_625k-MCSystem OBJECT IDENTIFIER
-- DESCRIPTION "System Elements"
```

```

 ::= { IEEE802dot20-625k-MC-MIB 1 }

_625k-MCSysAlarms                                OBJECT IDENTIFIER
  -- DESCRIPTION      "Alarms"
  ::= { _625k-MCSystem 1 }

_625k-MCAAlarmScalars                            OBJECT IDENTIFIER
  -- DESCRIPTION      "Alarm Scalars"
  ::= { _625k-MCSysAlarms 1 }

_625k-MCCommonAlarmStatus                        OBJECT-TYPE
  SYNTAX               INTEGER -- Unsigned32Type
  ACCESS               read-only
  STATUS               mandatory
  DESCRIPTION
    "Common alarm atatus.

    (From mibCtl ElementType 16 CommonAlarmStatus)
    Description for mibCtl Type 14 Unsigned32Type :
      32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
      32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
  ::= { _625k-MCAAlarmScalars 1 }

_625k-MCFailReasonForAlarm                      OBJECT-TYPE
  SYNTAX               INTEGER -- Unsigned32Type
  ACCESS               read-only
  STATUS               mandatory
  DESCRIPTION
    "Fail reason for alarm.

    (From mibCtl ElementType 15 FailReasonForAlarm)
    Description for mibCtl Type 14 Unsigned32Type :
      32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
      32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
  ::= { _625k-MCAAlarmScalars 2 }

_625k-MCAAlarmSummaryTable                      OBJECT-TYPE
  SYNTAX SEQUENCE OF _625k-MCAAlarmSummaryTableEntry
  ACCESS               not-accessible
  STATUS               mandatory
  DESCRIPTION          "Alarm Summary Table"
  ::= { _625k-MCSysAlarms 2 }

_625k-MCAAlarmSummaryTableEntry                OBJECT-TYPE
  SYNTAX               _625k-MCAAlarmSummaryTableEntry
  ACCESS               not-accessible
  STATUS               mandatory
  DESCRIPTION          ""
  INDEX { _625k-MCAAlarmSummaryTableIndex }

```

```

 ::= { _625k-MCAAlarmSummaryTable 1 }

_625k-MCAAlarmSummaryTableEntry ::= SEQUENCE {
    _625k-MCAAlarmSummaryTableIndex      INTEGER, -- AlarmEventType
    _625k-MCAAlarmSummary                INTEGER -- AlarmStateType
}

_625k-MCAAlarmSummaryTableIndex          OBJECT-TYPE
SYNTAX      INTEGER -- AlarmEventType
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "
        Description for mibCtl Type 85 AlarmEventType :
        Enumeration of alarm event types.

        Defines semantics of events that are also alarms.
        All alarm events are enumerated first in the list of event types.
        The highest alarm event index will never be more than 255.
        [Limits: 0 255 ]
        Type derived from mibCtl Type 3 EventType :
        Enumeration of event types.

        Defines semantics of events.
        An event is re. an event log message.
        [Limits: 0 255 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
 ::= { _625k-MCAAlarmSummaryTableEntry 1 }

_625k-MCAAlarmSummary                    OBJECT-TYPE
SYNTAX      INTEGER -- AlarmStateType
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "Summary of all alarms generated by the base station.

    Each element contains the summary of a type of alarm (e.g. module
    over temperature). When the management station sees that alarm
    summary is SET, it can, for example, query AlarmModuleOverTemp
    to see which module(s) is over temperature.

    (From mibCtl ElementType 5210 AlarmSummary)
    Description for mibCtl Type 80 AlarmStateType :
        Current state of an alarm.

        This value is CLEARED when
        the conditions which caused the alarm to occur are taken care of
        and no longer exist.
        The value is SET when due to some conditions, the Base Station
        software decides that an alarm is necessary.
        Typically (though this may not be true for all alarms
        or if the alarm changes state too frequently)
        an event is logged when an alarm is SET and then again when it
        is CLEARED.
        [Limits: 0 1 ]
        Description for mibCtl AlarmStateType 0 CLEARED :
        No alarm.
        Description for mibCtl AlarmStateType 1 SET :
        Alarm is set.
    "

```

```

 ::= { _625k-MCAAlarmSummaryTableEntry 2 }

_625k-MCSysFiles                                OBJECT IDENTIFIER
  -- DESCRIPTION                                "Files"
  ::= { _625k-MCSystem 2 }

_625k-MCStatsFiles                              OBJECT IDENTIFIER
  -- DESCRIPTION                                "Statistics file"
  ::= { _625k-MCSysFiles 1 }

_625k-MCStatsUploadURL                          OBJECT-TYPE
  SYNTAX OCTET STRING (SIZE(0..64)) -- URLType
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
    "EMS location to upload BS statistics file.

    (From mibCtl ElementType 2831 StatsUploadURL)
    Description for mibCtl Type 401 URLType :
      Universal Resource Locator (URL).

    A Universal Resource Locator (URL) is a text string
    that specifies a network location for a file.
    The general format for a URL consists of 2 parts:

    1. Protocol name: lower case letters, followed by a colon.
    See below for supported protocols.
    This field may be omitted, to default to the file: protocol.

    2. Additional information, depending on the protocol.
    For many protocols, a host name is required,
    which consists of a dotted numerical Internet Protocol (IP)
    address,
    or a dotted symbolic name with alphanumerical components,
    where supported.

    Supported protocols are:

    tftp: is the Trivial File Transfer Protocol.
    The additional information should begin with two slashes (//)
    followed by a host name, a slash (/) and a file path.
    The file path is interpreted by the host system,
    frequently relative to a special directory set up for this
    purpose.

    file: is the plain old file protocol.
    The additional information consists of a file path, which
    should begin with a slash (/).
    This is only useful if Base Station has been configured
    to be an Network File System (NFS) client of the host.
    The file path is interpreted on the Base Station, so it
    shall begin with the mount name specified in the NFS
    configuration.
    Type derived from mibCtl Type 15 TextType :
      ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
      8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
  ::= { _625k-MCStatsFiles 1 }

```

```

_625k-MCStatsUploadStatus                                OBJECT-TYPE
    SYNTAX          INTEGER -- FileUploadStatusType
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "Stats file upload status.

        (From mibCtl ElementType 2832 StatsUploadStatus)
        Description for mibCtl Type 403 FileUploadStatusType :
            File upload status.

        Description for mibCtl FileUploadStatusType 0 Unknown :
            File upload status is unknown.
        Description for mibCtl FileUploadStatusType 1 Missing :
            File is missing or invalid.
        Description for mibCtl FileUploadStatusType 2 PartialUpload :
            File is in the process of being upload to EMS.
        Description for mibCtl FileUploadStatusType 4 Complete :
            File is completely uploaded to EMS.
        Description for mibCtl FileUploadStatusType 5 Failure :
            Upload process is failure.
        Description for mibCtl FileUploadStatusType 6 NotManaged :
            File upload is not being managed.
        "
    ::= { _625k-MCStatsFiles 2 }

_625k-MCStatsUploadFailReason                            OBJECT-TYPE
    SYNTAX          INTEGER -- FileUploadFailReasonType
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "Reason for last stats file upload failure.

        (From mibCtl ElementType 2833 StatsUploadFailReason)
        Description for mibCtl Type 406 FileUploadFailReasonType :
            Reason for failure to upload a file..

        Description for mibCtl FileUploadFailReasonType 0 NoFailure :
            File upload in progress or completed without problem.
        Description for mibCtl FileUploadFailReasonType 1 BadPathSpecified :
            File upload failed because network path not found.
        Description for mibCtl FileUploadFailReasonType 2 FlashDiskReadError
        :
            File upload failed because of flash disk read error.
        Description for mibCtl FileUploadFailReasonType 3 Aborted :
            File upload aborted due to change of specification.
        Description for mibCtl FileUploadFailReasonType 4 WriteError :
            Error in putting a file.
        "
    ::= { _625k-MCStatsFiles 3 }

_625k-MCStatsUploadBytes                                OBJECT-TYPE
    SYNTAX          INTEGER -- Unsigned32Type
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION

```


"Upload size of BS stats file in bytes.

(From mibCtl ElementType 2834 StatsUploadBytes)
Description for mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
::= { _625k-MCStatsFiles 4 }

_625k-MCStatsUploadDate OBJECT-TYPE
SYNTAX Gauge -- AbsoluteTimeType
ACCESS read-only
STATUS mandatory
DESCRIPTION
"BS Stats File upload complete time.

(From mibCtl ElementType 2835 StatsUploadDate)
Description for mibCtl Type 801 AbsoluteTimeType :
Absolute time in GPS seconds.

1980. GPS (Global Positioning System) time in seconds since Jan. 6,

Note that this differs from UTC (in addition to a possible offset due to starting time) due to leap seconds; see the GpsLeapSecond element.
Type derived from mibCtl Type 18 Gauge32Type :
32 bits of Gauge data.
Derived from basic 32 bit word type.
"
::= { _625k-MCStatsFiles 5 }

_625k-MCUploadStatsFile OBJECT-TYPE
SYNTAX INTEGER -- Unsigned32Type
ACCESS read-write -- REALLY: write-only
STATUS mandatory
DESCRIPTION
"Upload Stats file.

(From mibCtl ElementType 2836 UploadStatsFile)
Description for mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
::= { _625k-MCStatsFiles 6 }

_625k-MCSysInterfaces OBJECT IDENTIFIER
-- DESCRIPTION "System Interfaces"
::= { _625k-MCSystem 3 }

_625k-MCInterfaceNetwork OBJECT IDENTIFIER
-- DESCRIPTION "Network Interfaces"

```
::= { _625k-MCSysInterfaces 1 }
```

```
_625k-MCTypeOfNetworkProtocol          OBJECT-TYPE
SYNTAX          INTEGER -- NetworkProtocolType
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Type of Network Protocol used with the Network.

    Type of Network Protocol is Ethernet or ATM.

    (From mibCtl ElementType 3002 TypeOfNetworkProtocol)
    Description for mibCtl Type 214 NetworkProtocolType :
        Network Protocol type.
    Description for mibCtl NetworkProtocolType 0 Unknown :
        Network protocol type is unknown.
    Description for mibCtl NetworkProtocolType 1 Ethernet :
        Ethernet interface.
    Description for mibCtl NetworkProtocolType 2 ATM :
        ATM interface.
    "
::= { _625k-MCInterfaceNetwork 1 }
```

```
_625k-MCMgmtNetConfigTable             OBJECT-TYPE
SYNTAX SEQUENCE OF _625k-MCMgmtNetConfigTableEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     "Mgmt Network Configuration"
::= { _625k-MCInterfaceNetwork 2 }
```

```
_625k-MCMgmtNetConfigTableEntry        OBJECT-TYPE
SYNTAX          _625k-MCMgmtNetConfigTableEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     ""
INDEX { _625k-MCMgmtNetConfigTableIndex }
::= { _625k-MCMgmtNetConfigTable 1 }

_625k-MCMgmtNetConfigTableEntry ::= SEQUENCE {
    _625k-MCMgmtNetConfigTableIndex    INTEGER, -- MoNerdAddressType
    _625k-MCEthernetIPAddress          OCTET STRING (SIZE(0..15)), --
    IPAddressTextType                  OCTET STRING (SIZE(0..15)), --
    _625k-MCEthernetIPLocalBits        OCTET STRING (SIZE(0..20)) --
    _625k-MCEthernetHostName
    TextType X 20
}
```

```
_625k-MCMgmtNetConfigTableIndex        OBJECT-TYPE
SYNTAX          INTEGER -- MoNerdAddressType
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "
    Description for mibCtl Type 204 MoNerdAddressType :
        Base station network component address.

    A network address is a subset of Base Station component addresses,
    restricted to network components only.
    Network components interface with a telephony switch or similar.
```

```

        [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { _625k-MCMgmtNetConfigTableEntry 1 }

_625k-MCEthernetIPAddress OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Internet Protocol (IP) address for ethernet port of Module.

        This is the actual IP address in use for the ethernet port
        of a given Module.
        If IP is not being used on the ethernet port, or there is
        no ethernet port, then an empty string is provided for this element.

        (From mibCtl ElementType 2811 EthernetIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

        This text shall currently be in the dotted abc.def.ghi.jkl
format.
        In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
    "
    ::= { _625k-MCMgmtNetConfigTableEntry 2 }

_625k-MCEthernetIPLocalBits OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Ethernet IP (Internet Protocol) local routing bit count.

        This indicates how many of the low-order bits of
        the IP address of the ethernet connection are used
        within the local network.
        The remaining (high-order) bits are the same for all
        hosts on the local network.
        This is used as the first part of the routing algorithm.
        IP addresses that do not share the upper bits of the ethernet
        IP address and which are not otherwise resolved will be sent
        through the gateway, if defined.

        For example, 255.255.255.0

        (From mibCtl ElementType 2812 EthernetIPLocalBits)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

        This text shall currently be in the dotted abc.def.ghi.jkl
format.
        In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.

```

```

Type derived from mibCtl Type 12 OctetType :
    8 bits of raw opaque data.
Derived from basic 8 bit word type.
"
 ::= { _625k-MCMgmtNetConfigTableEntry 3 }

```

```

_625k-MCEthernetHostName OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Ethernet IP host name for module.

        (From mibCtl ElementType 2813 EthernetHostName)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { _625k-MCMgmtNetConfigTableEntry 4 }

```

```

_625k-MCUserNetConfigTable OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCUserNetConfigTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "User Network Configuration"
    ::= { _625k-MCInterfaceNetwork 3 }

```

```

_625k-MCUserNetConfigTableEntry OBJECT-TYPE
    SYNTAX      _625k-MCUserNetConfigTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        ""
    INDEX { _625k-MCUserNetConfigTableIndex }
    ::= { _625k-MCUserNetConfigTable 1 }

```

```

_625k-MCUserNetConfigTableEntry ::= SEQUENCE {
    _625k-MCUserNetConfigTableIndex INTEGER, -- MoNerdAddressType
    _625k-MCUserEthernetIPAddress OCTET STRING (SIZE(0..15)), --
    IPAddressTextType
    _625k-MCUserEthernetIPLocalBits OCTET STRING (SIZE(0..15)), --
    IPAddressTextType
    _625k-MCUserEthernetHostName OCTET STRING (SIZE(0..20)) --
    TextType X 20
}

```

```

_625k-MCUserNetConfigTableIndex OBJECT-TYPE
    SYNTAX      INTEGER -- MoNerdAddressType
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

        A network address is a subset of Base Station component addresses,
        restricted to network components only.
        Network components interface with a telephony switch or similar.
        [Limits: 0 1 ]

```

```

        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { _625k-MCUserNetConfigTableEntry 1 }

_625k-MCUserEthernetIPAddress OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Internet Protocol (IP) address for user ethernet port of Module.

        This is the actual IP address in use for the ethernet port
        of a given Module.
        If IP is not being used on the ethernet port, or there is
        no ethernet port, then an empty string is provided for this element.

        (From mibCtl ElementType 2817 UserEthernetIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

        This text shall currently be in the dotted abc.def.ghi.jkl
format.
        In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
    "
    ::= { _625k-MCUserNetConfigTableEntry 2 }

_625k-MCUserEthernetIPLocalBits OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Ethernet IP subnet mask for user network.

        Ethernet IP subnet mask for user network

        (From mibCtl ElementType 2818 UserEthernetIPLocalBits)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

        This text shall currently be in the dotted abc.def.ghi.jkl
format.
        In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
    "
    ::= { _625k-MCUserNetConfigTableEntry 3 }

_625k-MCUserEthernetHostName OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
    ACCESS      read-write
    STATUS      mandatory

```

DESCRIPTION

"User ethernet IP host name for module.

(From mibCtl ElementType 2819 UserEthernetHostName)

Description for mibCtl Type 15 TextType :

ASCII or compatible text.

Type derived from mibCtl Type 12 OctetType :

8 bits of raw opaque data.

Derived from basic 8 bit word type.

"

::= { _625k-MCUserNetConfigTableEntry 4 }

```
_625k-MCUserNetStatusTable          OBJECT-TYPE
SYNTAX SEQUENCE OF _625k-MCUserNetStatusTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Network Status"
::= { _625K-MCInterfaceNetwork 4 }
```

```
_625k-MCUserNetStatusTableEntry      OBJECT-TYPE
SYNTAX _625k-MCUserNetStatusTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION ""
INDEX { _625k-MCUserNetStatusTableIndex }
::= { _625k-MCUserNetStatusTable 1 }
```

```
_625k-MCUserNetStatusTableEntry ::= SEQUENCE {
    _625k-MCUserNetStatusTableIndex    INTEGER, -- MoNerdAddressType
    _625k-MCNetworkInOctets            Counter, -- Counter32Type
    _625k-MCNetworkOutOctets           Counter -- Counter32Type
}
```

```
_625k-MCUserNetStatusTableIndex      OBJECT-TYPE
SYNTAX INTEGER -- MoNerdAddressType
ACCESS read-only
STATUS mandatory
DESCRIPTION "
    Description for mibCtl Type 204 MoNerdAddressType :
        Base station network component address.

    A network address is a subset of Base Station component addresses,
    restricted to network components only.
    Network components interface with a telephony switch or similar.
    [Limits: 0 1 ]
    Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
"
::= { _625k-MCUserNetStatusTableEntry 1 }
```

```
_625k-MCNetworkInOctets              OBJECT-TYPE
SYNTAX Counter -- Counter32Type
ACCESS read-only
STATUS mandatory
DESCRIPTION "In octets user data of network."
```

```

        (From mibCtl ElementType 1000 NetworkInOctets)
        Description for mibCtl Type 19 Counter32Type :
            32 bits of Counter data.
        Derived from basic 32 bit word type.
        "
    ::= { _625k-MCUserNetStatusTableEntry 2 }

```

```

_625k-MCNetworkOutOctets                                OBJECT-TYPE
    SYNTAX          Counter -- Counter32Type
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "Out octets user data of network.

```

```

        (From mibCtl ElementType 1001 NetworkOutOctets)
        Description for mibCtl Type 19 Counter32Type :
            32 bits of Counter data.
        Derived from basic 32 bit word type.
        "
    ::= { _625k-MCUserNetStatusTableEntry 3 }

```

```

_625k-MCL2TPConfigTable                                OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCL2TPConfigTableEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION
        "L2TP Configuration Table"
    ::= { _625k-MCInterfaceNetwork 5 }

```

```

_625k-MCL2TPConfigTableEntry                            OBJECT-TYPE
    SYNTAX          _625k-MCL2TPConfigTableEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION
        ""
    INDEX { _625k-MCL2TPConfigTableIndex }
    ::= { _625k-MCL2TPConfigTable 1 }

```

```

_625k-MCL2TPConfigTableEntry ::= SEQUENCE {
    _625k-MCL2TPConfigTableIndex    INTEGER, -- MoNerdAddressType
    _625k-MCL2TPPeerName            OCTET STRING (SIZE(0..20)), --
TextType X 20
    _625k-MCL2TPPeerIPAddress       OCTET STRING (SIZE(0..15)), --
IPAddressTextType
    _625k-MCL2TPAVPHostName         OCTET STRING (SIZE(0..20)), --
TextType X 20
    _625k-MCL2TPAVPChallAndRes      OCTET STRING (SIZE(0..20)) --
TextType X 20
}

```

```

_625k-MCL2TPConfigTableIndex                            OBJECT-TYPE
    SYNTAX          INTEGER -- MoNerdAddressType
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "
        Description for mibCtl Type 204 MoNerdAddressType :

```

Base station network component address.

A network address is a subset of Base Station component addresses, restricted to network components only.

Network components interface with a telephony switch or similar.
[Limits: 0 1]

Type derived from mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.

Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.

Derived from basic 32 bit word type.

"
::= { _625k-MCL2TPConfigTableEntry 1 }

_625k-MCL2TPPeerName OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS read-write
STATUS mandatory
DESCRIPTION
"L2TP peer name.

Tunnel switch host name

(From mibCtl ElementType 2000 L2TPPeerName)
Description for mibCtl Type 15 TextType :
ASCII or compatible text.
Type derived from mibCtl Type 12 OctetType :
8 bits of raw opaque data.
Derived from basic 8 bit word type.
"
::= { _625k-MCL2TPConfigTableEntry 2 }

_625k-MCL2TPPeerIPAddress OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..15)) -- IPAddressTextType
ACCESS read-write
STATUS mandatory
DESCRIPTION
"L2TP peer IP Address.

Tunnel switch IP Address

(From mibCtl ElementType 2001 L2TPPeerIPAddress)
Description for mibCtl Type 420 IPAddressTextType :
Internet Protocol Address (Text).

This text shall currently be in the dotted abc.def.ghi.jkl
format.

In the future, hostnames might be allowed.

Type derived from mibCtl Type 15 TextType :
ASCII or compatible text.

Type derived from mibCtl Type 12 OctetType :
8 bits of raw opaque data.

Derived from basic 8 bit word type.

"
::= { _625k-MCL2TPConfigTableEntry 3 }

_625k-MCL2TPAVPHostName OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS read-write
STATUS mandatory
DESCRIPTION
"BS host name using L2TP.


```

        (From mibCtl ElementType 2008 L2TPAVPHostName)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { _625k-MCL2TPConfigTableEntry 4 }

_625k-MCL2TPAVPChallAndRes OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "AVP challenge and response name.

        (From mibCtl ElementType 2012 L2TPAVPChallAndRes)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { _625k-MCL2TPConfigTableEntry 5 }

_625k-MCL2TPStatusTable OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCL2TPStatusTableEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION "L2TP Status Table"
    ::= { _625k-MCInterfaceNetwork 6 }

_625k-MCL2TPStatusTableEntry OBJECT-TYPE
    SYNTAX      _625k-MCL2TPStatusTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION ""
    INDEX { _625k-MCL2TPStatusTableIndex }
    ::= { _625k-MCL2TPStatusTable 1 }

_625k-MCL2TPStatusTableEntry ::= SEQUENCE {
    _625k-MCL2TPStatusTableIndex INTEGER, -- MoNerdAddressType
    _625k-MCL2TPActiveSession   INTEGER, -- Unsigned32Type
    _625k-MCL2TPActiveTunnel    INTEGER -- Unsigned32Type
}

_625k-MCL2TPStatusTableIndex OBJECT-TYPE
    SYNTAX      INTEGER -- MoNerdAddressType
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

        A network address is a subset of Base Station component addresses,

```

```

        restricted to network components only.
        Network components interface with a telephony switch or similar.
        [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { _625k-MCL2TPStatusTableEntry 1 }

_625k-MCL2TPActiveSession                                OBJECT-TYPE
    SYNTAX          INTEGER -- Unsigned32Type
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "L2TP active session.

        (From mibCtl ElementType 2013 L2TPActiveSession)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { _625k-MCL2TPStatusTableEntry 2 }

_625k-MCL2TPActiveTunnel                                OBJECT-TYPE
    SYNTAX          INTEGER -- Unsigned32Type
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "L2TP active tunnel.

        (From mibCtl ElementType 2014 L2TPActiveTunnel)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { _625k-MCL2TPStatusTableEntry 3 }

_625k-MCATMConfigTable                                OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCATMConfigTableEntry
    ACCESS           not-accessible
    STATUS           mandatory
    DESCRIPTION
        "ATM Configuration Table"
    ::= { _625k-MCInterfaceNetwork 7 }

_625k-MCATMConfigTableEntry                            OBJECT-TYPE
    SYNTAX          _625k-MCATMConfigTableEntry
    ACCESS           not-accessible
    STATUS           mandatory
    DESCRIPTION
        " "
```

```

INDEX { _625k-MCATMConfigTableIndex }
 ::= { _625k-MCATMConfigTable 1 }

_625k-MCATMConfigTableEntry ::= SEQUENCE {
  _625k-MCATMConfigTableIndex      INTEGER, -- MoNerdAddressType
  _625k-MCATmAddress                OCTET STRING (SIZE(0..40)), --
  TextType X 40
  _625k-MCATmVCTypes                INTEGER, -- AtmVCType
  _625k-MCATmFrameTypes              INTEGER, -- AtmFrameType
  _625k-MCATmUNIVersion              INTEGER, -- AtmUNIVersionType
  _625k-MCATmLineStatus              INTEGER, -- LineStatusType
  _625k-MCATmParameterFailReason     INTEGER, -- Unsigned32Type
  _625k-MCATmOpenChannelFailReason   INTEGER, -- Unsigned32Type
  _625k-MCATmChannelNumber           INTEGER, -- Unsigned32Type
  _625k-MCATmAlarmCauseRegister      INTEGER, -- Unsigned32Type
  _625k-MCATmPHYIntrCauseRegister    INTEGER -- Unsigned32Type
}

_625k-MCATMConfigTableIndex      OBJECT-TYPE
SYNTAX      INTEGER -- MoNerdAddressType
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "
    Description for mibCtl Type 204 MoNerdAddressType :
    Base station network component address.

    A network address is a subset of Base Station component addresses,
    restricted to network components only.
    Network components interface with a telephony switch or similar.
    [Limits: 0 1 ]
    Type derived from mibCtl Type 14 Unsigned32Type :
    32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
    32 bits of raw opaque data.
    Derived from basic 32 bit word type.
"
 ::= { _625k-MCATMConfigTableEntry 1 }

_625k-MCATmAddress                OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..40)) -- TextType X 40
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "Atm Address.

    (From mibCtl ElementType 1950 AtmAddress)
    Description for mibCtl Type 15 TextType :
    ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
    8 bits of raw opaque data.
    Derived from basic 8 bit word type.
"
 ::= { _625k-MCATMConfigTableEntry 2 }

_625k-MCATmVCTypes                OBJECT-TYPE
SYNTAX      INTEGER -- AtmVCType
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "Atm VC Type.

```

```
(From mibCtl ElementType 1951 AtmVCTypes)
Description for mibCtl Type 300 AtmVCType :
    ATM VC Type.
Description for mibCtl AtmVCType 0 Unknown :
    ATM VC Type is Unknown.
Description for mibCtl AtmVCType 1 PVC :
    ATM VC Type is PVC.
Description for mibCtl AtmVCType 2 SVC :
    ATM VC Type is SVC.
Description for mibCtl AtmVCType 3 PVC SVC :
    ATM VC Type is PVC & SVC.
"
::= { _625k-MCATMConfigTableEntry 3 }
```

```
_625k-MCATmFrameTypes                                OBJECT-TYPE
SYNTAX          INTEGER -- AtmFrameType
ACCESS          read-write
STATUS          mandatory
DESCRIPTION     "Atm Frame Type."
```

```
(From mibCtl ElementType 1952 AtmFrameTypes)
Description for mibCtl Type 301 AtmFrameType :
    ATM Frame Type.
Description for mibCtl AtmFrameType 0 Unknown :
    ATM Frame Type is Unknown.
Description for mibCtl AtmFrameType 1 OC48 :
    ATM Frame Type is OC48.
Description for mibCtl AtmFrameType 2 OC36 :
    ATM Frame Type is OC36.
Description for mibCtl AtmFrameType 3 OC24 :
    ATM Frame Type is OC24.
Description for mibCtl AtmFrameType 4 OC18 :
    ATM Frame Type is OC18.
Description for mibCtl AtmFrameType 5 OC12 :
    ATM Frame Type is OC12.
Description for mibCtl AtmFrameType 6 OC9 :
    ATM Frame Type is OC9.
Description for mibCtl AtmFrameType 7 OC3 :
    ATM Frame Type is OC3.
Description for mibCtl AtmFrameType 8 OC1 :
    ATM Frame Type is OC1.
Description for mibCtl AtmFrameType 9 STM16 :
    ATM Frame Type is STM16.
Description for mibCtl AtmFrameType 10 STM4 :
    ATM Frame Type is STM4.
Description for mibCtl AtmFrameType 11 STM1 :
    ATM Frame Type is STM1.
Description for mibCtl AtmFrameType 12 DS3 :
    ATM Frame Type is DS3.
Description for mibCtl AtmFrameType 13 DS2 :
    ATM Frame Type is DS2.
Description for mibCtl AtmFrameType 14 DS1 :
    ATM Frame Type is DS1.
Description for mibCtl AtmFrameType 15 DS0 :
    ATM Frame Type is DS0.
Description for mibCtl AtmFrameType 16 E3 :
    ATM Frame Type is E3.
Description for mibCtl AtmFrameType 17 E2 :
    ATM Frame Type is E2.
Description for mibCtl AtmFrameType 18 E1 :
```

```

        ATM Frame Type is E1.
        Description for mibCtl AtmFrameType 19 E0 :
            ATM Frame Type is E0.
    "
 ::= { _625k-MCATMConfigTableEntry 4 }

```

```

_625k-MCAtmUNIVersion                                OBJECT-TYPE
    SYNTAX      INTEGER -- AtmUNIVersionType
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Atm UNI version.

        (From mibCtl ElementType 1953 AtmUNIVersion)
        Description for mibCtl Type 302 AtmUNIVersionType :
            ATM UNI Version Type.
        Description for mibCtl AtmUNIVersionType 0 Unknown :
            ATM UNI Version Type is Unknown.
        Description for mibCtl AtmUNIVersionType 1 V30 :
            ATM UNI Version Type is 3.0.
        Description for mibCtl AtmUNIVersionType 2 V31 :
            ATM UNI Version Type is 3.1.
        Description for mibCtl AtmUNIVersionType 3 V40 :
            ATM UNI Version Type is 4.0.
    "
 ::= { _625k-MCATMConfigTableEntry 5 }

```

```

_625k-MCAtmLineStatus                                OBJECT-TYPE
    SYNTAX      INTEGER -- LineStatusType
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Atm line status.

        ATM line status

        (From mibCtl ElementType 1956 AtmLineStatus)
        Description for mibCtl Type 72 LineStatusType :
            Line status type.
        Description for mibCtl LineStatusType 0 LinkUp :
            Line status is link up.
        Description for mibCtl LineStatusType 1 LinkDown :
            Line status is link down.
    "
 ::= { _625k-MCATMConfigTableEntry 6 }

```

```

_625k-MCAtmParameterFailReason                        OBJECT-TYPE
    SYNTAX      INTEGER -- Unsigned32Type
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Atm parameter fail reason.

        ATM parameter fail reason

        (From mibCtl ElementType 1957 AtmParameterFailReason)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.

```

Derived from basic 32 bit word type.
"
::= { _625k-MCATMConfigTableEntry 7 }

_625k-MCATmOpenChannelFailReason OBJECT-TYPE
SYNTAX INTEGER -- Unsigned32Type
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Atm open channel fail reason.

ATM open channel fail reason

(From mibCtl ElementType 1958 AtmOpenChannelFailReason)
Description for mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
::= { _625k-MCATMConfigTableEntry 8 }

_625k-MCATmChannelNumber OBJECT-TYPE
SYNTAX INTEGER -- Unsigned32Type
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Atm open channel fail reason.

ATM open channel fail reason

(From mibCtl ElementType 1959 AtmChannelNumber)
Description for mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
::= { _625k-MCATMConfigTableEntry 9 }

_625k-MCATmAlarmCauseRegister OBJECT-TYPE
SYNTAX INTEGER -- Unsigned32Type
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Atm alarm cause register.

ATM alarm cause register

(From mibCtl ElementType 1960 AtmAlarmCauseRegister)
Description for mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
::= { _625k-MCATMConfigTableEntry 10 }

_625k-MCATmPHYIntrCauseRegister OBJECT-TYPE
SYNTAX INTEGER -- Unsigned32Type
ACCESS read-only

```

STATUS          mandatory
DESCRIPTION
    "Atm PHY Interrupt cause register.

    ATM PHY interrupt cause register

    (From mibCtl ElementType 1961 AtmPHYIntrCauseRegister)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
::= { _625k-MCATMConfigTableEntry 11 }

_625k-MCA10ConfigTable          OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCA10ConfigTableEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION "A10 Configuration Table"
    ::= { _625k-MCInterfaceNetwork 9 }

_625k-MCA10ConfigTableEntry     OBJECT-TYPE
    SYNTAX _625k-MCA10ConfigTableEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION ""
    INDEX { _625k-MCA10ConfigTableIndex }
    ::= { _625k-MCA10ConfigTable 1 }

_625k-MCA10ConfigTableEntry ::= SEQUENCE {
    _625k-MCA10ConfigTableIndex      INTEGER, -- MoNerdAddressType
    _625k-MC3GPP2PriPDSNIPAddress    OCTET STRING (SIZE(0..15)), --
    IPAddressTextType                 OCTET STRING (SIZE(0..64)), --
    _625k-MC3GPP2PriPDSNSharedSecret TextType X 64
    _625k-MC3GPP2PriPDSNSPI           INTEGER, -- Unsigned32Type
    _625k-MC3GPP2SecPDSNIPAddress     OCTET STRING (SIZE(0..15)), --
    IPAddressTextType                 OCTET STRING (SIZE(0..64)), --
    _625k-MC3GPP2SecPDSNSharedSecret TextType X 64
    _625k-MC3GPP2SecPDSNSPI           INTEGER -- Unsigned32Type
}

_625k-MCA10ConfigTableIndex     OBJECT-TYPE
    SYNTAX INTEGER -- MoNerdAddressType
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "

```

```

"
 ::= { _625k-MCA10ConfigTableEntry 1 }

_625k-MC3GPP2PriPDSNIPAddress          OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "3GPP2 primary PDSN IP address.

    3GPP2 Primary PDSN IP Address

    (From mibCtl ElementType 2100 3GPP2PriPDSNIPAddress)
    Description for mibCtl Type 420 IPAddressTextType :
        Internet Protocol Address (Text).

    This text shall currently be in the dotted abc.def.ghi.jkl
format.
    In the future, hostnames might be allowed.
    Type derived from mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
"
 ::= { _625k-MCA10ConfigTableEntry 2 }

_625k-MC3GPP2PriPDSNSharedSecret        OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..64)) -- TextType X 64
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "3GPP2 primary PDSN shared secret.

    3GPP2 primary PDSN shared secret

    (From mibCtl ElementType 2101 3GPP2PriPDSNSharedSecret)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
"
 ::= { _625k-MCA10ConfigTableEntry 3 }

_625k-MC3GPP2PriPDSNSPI                 OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "3GPP2 primary PDSN SPI.

    3GPP2 primary PDSN SPI

    (From mibCtl ElementType 2102 3GPP2PriPDSNSPI)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
"
 ::= { _625k-MCA10ConfigTableEntry 4 }

```



```

_625k-MC3GPP2SecPDSNIPAddress          OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "3GPP2 secondary PDSN IP address.

    3GPP2 secondary PDSN IP address

    (From mibCtl ElementType 2103 3GPP2SecPDSNIPAddress)
    Description for mibCtl Type 420 IPAddressTextType :
        Internet Protocol Address (Text).

        This text shall currently be in the dotted abc.def.ghi.jkl
format.
        In the future, hostnames might be allowed.
    Type derived from mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
 ::= { _625k-MCA10ConfigTableEntry 5 }

_625k-MC3GPP2SecPDSNSharedSecret        OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..64)) -- TextType X 64
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "3GPP2 secondary PDSN shared secret.

    3GPP2 secondary PDSN shared secret

    (From mibCtl ElementType 2104 3GPP2SecPDSNSharedSecret)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
 ::= { _625k-MCA10ConfigTableEntry 6 }

_625k-MC3GPP2SecPDSNSPI                 OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "3GPP2 secondary PDSN SPI.

    3GPP2 secondary PDSN SPI

    (From mibCtl ElementType 2105 3GPP2SecPDSNSPI)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
 ::= { _625k-MCA10ConfigTableEntry 7 }

```

```

_625k-MCA10StatusTable                                OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCA10StatusTableEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION "A10 Status Table"
    ::= { _625k-MCInterfaceNetwork 10 }

_625k-MCA10StatusTableEntry                            OBJECT-TYPE
    SYNTAX _625k-MCA10StatusTableEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION ""
    INDEX { _625k-MCA10StatusTableIndex }
    ::= { _625k-MCA10StatusTable 1 }

_625k-MCA10StatusTableEntry ::= SEQUENCE {
    _625k-MCA10StatusTableIndex INTEGER, -- MoNerdAddressType
    _625k-MC3GPP2PDSNIPAddress OCTET STRING (SIZE(0..15)) --
    IPAddressTextType
}

_625k-MCA10StatusTableIndex                            OBJECT-TYPE
    SYNTAX INTEGER -- MoNerdAddressType
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "
        Description for mibCtl Type 204 MoNerdAddressType :
        Base station network component address.

        A network address is a subset of Base Station component addresses,
        restricted to network components only.
        Network components interface with a telephony switch or similar.
        [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { _625k-MCA10StatusTableEntry 1 }

_625k-MC3GPP2PDSNIPAddress                            OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "3GPP2 PDSN IP Address.

        Current main using PDSN IP address

        (From mibCtl ElementType 2113 3GPP2PDSNIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
        Internet Protocol Address (Text).

        This text shall currently be in the dotted abc.def.ghi.jkl
format.

        In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
        ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
    "
    ::= { _625k-MC3GPP2PDSNIPAddress 1 }

```

```

        8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
        ::= { _625k-MCA10StatusTableEntry 2 }

_625k-MCInterfaceRF                                OBJECT IDENTIFIER
-- DESCRIPTION "Radio Frequency Interfaces"
::= { _625k-MCSysInterfaces 2 }

_625k-MCCarrierTable                                OBJECT-TYPE
SYNTAX SEQUENCE OF _625k-MCCarrierTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Carrier Frequency Table"
::= { _625k-MCInterfaceRF 1 }

_625k-MCCarrierTableEntry                            OBJECT-TYPE
SYNTAX _625k-MCCarrierTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION ""
INDEX { _625k-MCCarrierTableIndex }
::= { _625k-MCCarrierTable 1 }

_625k-MCCarrierTableEntry ::= SEQUENCE {
    _625k-MCCarrierTableIndex INTEGER, -- BaseStationCarrierType
    _625k-MCCarrierUsage      INTEGER -- CarrierUsageType
}

_625k-MCCarrierTableIndex                            OBJECT-TYPE
SYNTAX INTEGER -- BaseStationCarrierType
ACCESS read-only
STATUS mandatory
DESCRIPTION "
    Description for mibCtl Type 219 BaseStationCarrierType :
    Base station carrier number.

    Base station carriers are a contiguous set of carriers
    that are used by the Base Station;
    they are numbered from 0 to a current maximum of 32-1.
    [Limits: 0 15 ]
    Type derived from mibCtl Type 14 Unsigned32Type :
    32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
    32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
    ::= { _625k-MCCarrierTableEntry 1 }

_625k-MCCarrierUsage                                OBJECT-TYPE
SYNTAX INTEGER -- CarrierUsageType
ACCESS read-only
STATUS mandatory
DESCRIPTION "Current assigned usage per base station carrier.

```

(From mibCtl ElementType 54 CarrierUsage)
 Description for mibCtl Type 220 CarrierUsageType :
 The assigned use of a radio carrier.

 A radio carrier is a frequency band.
 The assigned use of a carrier can be Reserved, Control
 or Traffic.
 Base station transmits control information on one of the time
 slots
 of given Control carrier.
 Base station does not transmit anything on Reserved carriers.

Description for mibCtl CarrierUsageType 0 NotUse :
 Not Use for this carrier.

Description for mibCtl CarrierUsageType 1 TCH :
 All timeslots in this carrier are for traffic only.

Description for mibCtl CarrierUsageType 2 TCHBCH :
 One timeslot in this carrier is for BCH, others for TCH.

"

::= { _625k-MCCarrierTableEntry 2 }

_625k-MCBSCC OBJECT-TYPE
 SYNTAX INTEGER -- Unsigned32Type
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "Base Station Color Code.

 (From mibCtl ElementType 61 BSCC)
 Description for mibCtl Type 14 Unsigned32Type :
 32 bit unsigned integer.
 Type derived from mibCtl Type 11 Word32Type :
 32 bits of raw opaque data.
 Derived from basic 32 bit word type.

"

::= { _625k-MCInterfaceRF 2 }

_625k-MCBSLowestCarrier OBJECT-TYPE
 SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "The lowest carrier of the base station operating band.

 This is an extended carrier number
 that identifies the lowest carrier of the bandwidth
 to which the base station is tuned.
 This value cannot be changed while the Base Station state
 is Operating.

 (From mibCtl ElementType 52 BSLowestCarrier)
 Description for mibCtl Type 15 TextType :
 ASCII or compatible text.
 Type derived from mibCtl Type 12 OctetType :
 8 bits of raw opaque data.
 Derived from basic 8 bit word type.

"

::= { _625k-MCInterfaceRF 3 }

_625k-MCBCHModuleAddress OBJECT-TYPE
 SYNTAX INTEGER -- ModuleAddressType
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION

"Which module is handling the broadcast channel.

(From mibCtl ElementType 57 BCHModuleAddress)
Description for mibCtl Type 202 ModuleAddressType :
Base station bus slot address.

Most components of the Base Station for which data can
be obtained are identified by a ModuleAddressType address
and possibly a subsidiary address.
[Limits: 0 7]

Type derived from mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.

Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.

Derived from basic 32 bit word type.

"

::= { _625k-MCInterfaceRF 4 }

_625k-MCBCHCarrierNumber OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Number Of BCH Carrier.

(From mibCtl ElementType 58 BCHCarrierNumber)
Description for mibCtl Type 15 TextType :
ASCII or compatible text.

Type derived from mibCtl Type 12 OctetType :
8 bits of raw opaque data.

Derived from basic 8 bit word type.

"

::= { _625k-MCInterfaceRF 5 }

_625k-MCRACHCarrierMask OBJECT-TYPE
SYNTAX INTEGER -- Unsigned32Type
ACCESS read-write
STATUS mandatory
DESCRIPTION
"RACH carrier mask.

(From mibCtl ElementType 73 RACHCarrierMask)
Description for mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.

Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.

Derived from basic 32 bit word type.

"

::= { _625k-MCInterfaceRF 6 }

_625k-MCRACHSlotMask OBJECT-TYPE
SYNTAX INTEGER -- Unsigned32Type
ACCESS read-write
STATUS mandatory
DESCRIPTION
"RACH slot mask.

(From mibCtl ElementType 72 RACHSlotMask)

```
Description for mibCtl Type 14 Unsigned32Type :
    32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
    32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
 ::= { _625k-MCInterfaceRF 7 }
```

```
_625k-MCCalibrationInterval          OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Calibration interval time.
```

```
(From mibCtl ElementType 75 CalibrationInterval)
Description for mibCtl Type 14 Unsigned32Type :
    32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
    32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
 ::= { _625k-MCInterfaceRF 8 }
```

```
_625k-MCSpatialParameter            OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Spatial parameter.
```

```
(From mibCtl ElementType 78 SpatialParameter)
Description for mibCtl Type 14 Unsigned32Type :
    32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
    32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
 ::= { _625k-MCInterfaceRF 9 }
```

```
_625k-MCCostCalcParameter            OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Cost calculation parameter.
```

```
(From mibCtl ElementType 79 CostCalcParameter)
Description for mibCtl Type 14 Unsigned32Type :
    32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
    32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
 ::= { _625k-MCInterfaceRF 10 }
```

```

_625k-MCBSRegistrationCapacity          OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Capacity of UT registration on BS.

    (From mibCtl ElementType 76 BSRegistrationCapacity)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
 ::= { _625k-MCInterfaceRF 11 }

_625k-MCBSRegistrationTimer              OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Timer of keeping UT registration on BS.

    (From mibCtl ElementType 77 BSRegistrationTimer)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
 ::= { _625k-MCInterfaceRF 12 }

_625k-MCPCHFrequencyHopping              OBJECT-TYPE
SYNTAX          INTEGER -- BooleanType
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Propriety of frequency hopping (PCH).

    (From mibCtl ElementType 70 PCHFrequencyHopping)
    Description for mibCtl Type 16 BooleanType :
        Truth value, 0=FALSE, 1=TRUE.

        This is a subset of TriStateType; no UNDEFINED value is provided.
        [Limits: 0 1 ]
    Description for mibCtl BooleanType 0 FALSE :
        False.
    Description for mibCtl BooleanType 1 TRUE :
        True.
    "
 ::= { _625k-MCInterfaceRF 13 }

_625k-MCTCHFrequencyHopping              OBJECT-TYPE
SYNTAX          INTEGER -- BooleanType

```

ACCESS read-write
STATUS mandatory
DESCRIPTION
"Propriety of frequency hopping (TCH)."

(From mibCtl ElementType 71 TCHFrequencyHopping)
Description for mibCtl Type 16 BooleanType :
Truth value, 0=FALSE, 1=TRUE.

This is a subset of TriStateType; no UNDEFINED value is provided.
[Limits: 0 1]
Description for mibCtl BooleanType 0 FALSE :
False.
Description for mibCtl BooleanType 1 TRUE :
True.

"
::= { _625k-MCInterfaceRF 14 }

_625k-MCRFStatusTable OBJECT-TYPE
SYNTAX SEQUENCE OF _625k-MCRFStatusTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "RF Status Table"
::= { _625k-MCInterfaceRF 15 }

_625k-MCRFStatusTableEntry OBJECT-TYPE
SYNTAX _625k-MCRFStatusTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION ""
INDEX { _625k-MCRFStatusTableIndex }
::= { _625k-MCRFStatusTable 1 }

_625k-MCRFStatusTableEntry ::= SEQUENCE {
_625k-MCRFStatusTableIndex INTEGER, -- MoNerdAddressType
_625k-MCBSAirBitRateUpLink INTEGER, -- Unsigned32Type
_625k-MCBSAirBitRateDownLink INTEGER, -- Unsigned32Type
_625k-MCBSActiveStream INTEGER, -- Unsigned32Type
_625k-MCBSActiveRegistration INTEGER -- Unsigned32Type
}

_625k-MCRFStatusTableIndex OBJECT-TYPE
SYNTAX INTEGER -- MoNerdAddressType
ACCESS read-only
STATUS mandatory
DESCRIPTION "
Description for mibCtl Type 204 MoNerdAddressType :
Base station network component address.

A network address is a subset of Base Station component addresses,
restricted to network components only.
Network components interface with a telephony switch or similar.
[Limits: 0 1]
Type derived from mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.
Derived from basic 32 bit word type.
"


```
::= { _625k-MCRFStatusTableEntry 1 }
```

```
_625k-MCBSAirBitRateUpLink          OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Radio bit rate of up link per Modem control board.
```

```

    (From mibCtl ElementType 4022 BSAirBitRateUpLink)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
```

```
::= { _625k-MCRFStatusTableEntry 2 }
```

```
_625k-MCBSAirBitRateDownLink        OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Radio bit rate of down link per Modem control board.
```

```

    (From mibCtl ElementType 4023 BSAirBitRateDownLink)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
```

```
::= { _625k-MCRFStatusTableEntry 3 }
```

```
_625k-MCBSActiveStream              OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Number of streams currently connected in a base station.
```

```

    Number of active streams.
```

```

    (From mibCtl ElementType 4020 BSActiveStream)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
```

```
::= { _625k-MCRFStatusTableEntry 4 }
```

```
_625k-MCBSActiveRegistration        OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
```

"Number of registrations currently existed in a base station.

The call capacity is determined by the available resources in a base station.

(From mibCtl ElementType 4021 BSAActiveRegistration)

Description for mibCtl Type 14 Unsigned32Type :

32 bit unsigned integer.

Type derived from mibCtl Type 11 Word32Type :

32 bits of raw opaque data.

Derived from basic 32 bit word type.

"

::= { _625k-MCRFStatusTableEntry 5 }

_625k-MCSysScalars OBJECT IDENTIFIER

-- DESCRIPTION "System Scalars"

::= { _625k-MCSystem 4 }

_625k-MCBaseStationID OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..18)) -- TextType X 18

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Base Station Identification Code.

This text string shall represent in hexadecimal a 42 bit number to be used as the Base Station Identification Code (BSID).

The BSID is used by the base station to identify itself to subscriber units.

The BSID of a base station shall at a minimum differ from that of any other base station where both would be within radio reception distance of any subscriber unit.

This cannot be changed while the Base Station state is Operating.

(From mibCtl ElementType 60 BaseStationID)

Description for mibCtl Type 15 TextType :

ASCII or compatible text.

Type derived from mibCtl Type 12 OctetType :

8 bits of raw opaque data.

Derived from basic 8 bit word type.

"

::= { _625k-MCSysScalars 1 }

_625k-MCBaseStationTypeID OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Type ID of base station.

(From mibCtl ElementType 66 BaseStationTypeID)

Description for mibCtl Type 15 TextType :

ASCII or compatible text.

Type derived from mibCtl Type 12 OctetType :

8 bits of raw opaque data.

Derived from basic 8 bit word type.

"

::= { _625k-MCSysScalars 2 }

```

_625k-MCBaseStationGroupID          OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Group ID of base station.

    (From mibCtl ElementType 67 BaseStationGroupID)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
 ::= { _625k-MCSysScalars 3 }

_625k-MCBaseStationSubGroupID        OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Sub group ID of base station.

    (From mibCtl ElementType 68 BaseStationSubGroupID)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
 ::= { _625k-MCSysScalars 4 }

_625k-MCDesiredStateOfBaseStation    OBJECT-TYPE
SYNTAX          INTEGER -- ComponentStateType
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Desired state of base station as a whole.

    This indicates the Base Station state desired by the operator.
    These desired states are currently supported:

    Operating -   for normal operation.

    Ready -      to avoid taking any new calls.
    Existing calls will not be terminated except normally or by
    command from the operator.
    While existing calls remain, the base station state will
    remain as Operating.

    This information is permanently stored on the base station.

    (From mibCtl ElementType 42 DesiredStateOfBaseStation)
    Description for mibCtl Type 71 ComponentStateType :
        Component operational state.

    A component begins in the Unknown state.
    If not detected, it enters and remains in the NotPresent state.
    If detected, it enters the Uninitialized state, from where it

```

may go to the Testing and Initializing states and then to the Standby or Operating state depending upon permissions.
Due to loss of permissions or resources, it may revert from the Operating state to the Standby state.
Due to failure or loss of permission, it may revert to the Uninitialized state, perhaps by way of the ShuttingDown state depending on the device.
From the Uninitialized state it may return to more advanced states depending upon permissions.
In case of a waiting period before (again) initializing, the component is considered to be Initializing.

Permissions include administrative permissions (from the operator); excessive failure restrictions; etc.
Description for mibCtl ComponentStateType 0 Unknown :
Component state not known.
Description for mibCtl ComponentStateType 1 NotPresent :
Component is not present.
Description for mibCtl ComponentStateType 2 PowerOff :
Component is present but powered off.
Description for mibCtl ComponentStateType 3 Uninitialized :
Component is present but not in use.

The power on/off state of the component is not specified in this case.
Description for mibCtl ComponentStateType 4 Testing :
Component is being tested.
Description for mibCtl ComponentStateType 5 Initializing :
Component is being initialized.
Description for mibCtl ComponentStateType 6 Ready :
Component is ready but not operating.
Description for mibCtl ComponentStateType 7 Operating :
Component is operating for normal use without restriction.

The component is either in actual use or may be used at any time, without restriction.
Description for mibCtl ComponentStateType 8 Abandoned :
Component state is not the desired state due to excessive errors.

The component state is not that desired, and the Base Station software has abandoned attempts to place the component in the desired state.
The actual state of the component is undefined.
The Base Station software will resume attempting to place the component in the desired state if the appropriate Reinitialize action element is written with the correct value.
Also, the software may resume attempts under other conditions, not all of which may be documented.
Description for mibCtl ComponentStateType 9 InitialSetUp :
Component is initial set up..

Initial set up state.
Description for mibCtl ComponentStateType 10 Degrading :
Component is degrading..

Degrading state.
Description for mibCtl ComponentStateType 11 Restriction :
Component is restriction..

Restriction state.

"
::= { _625k-MCSysScalars 5 }

_625k-MCTypeOfReboot	OBJECT-TYPE
SYNTAX	INTEGER -- RebootType
ACCESS	read-write

```

STATUS          mandatory
DESCRIPTION
    "Type of reboot for base station.

    (From mibCtl ElementType 63 TypeOfReboot)
    Description for mibCtl Type 250 RebootType :
        Reboot Type.
    Description for mibCtl RebootType 0 Force :
        Force mode.
    Description for mibCtl RebootType 1 Graceful :
        Graceful mode.
    "
::= { _625k-MCSysScalars 6 }

_625k-MCBaseStationRebootTime          OBJECT-TYPE
SYNTAX          Gauge -- AbsoluteTimeType
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Time of base station reboot.

    This is the base station reboot time (GPS time).

    (From mibCtl ElementType 62 BaseStationRebootTime)
    Description for mibCtl Type 801 AbsoluteTimeType :
        Absolute time in GPS seconds.

    GPS (Global Positioning System) time in seconds since Jan. 6,
1980.
    Note that this differs from UTC (in addition to a possible
    offset due to starting time) due to leap seconds; see
    the GpsLeapSecond element.
    Type derived from mibCtl Type 18 Gauge32Type :
        32 bits of Gauge data.
    Derived from basic 32 bit word type.
    "
::= { _625k-MCSysScalars 7 }

_625k-MCTypeOfBSDiagnosis              OBJECT-TYPE
SYNTAX          INTEGER -- DiagnosisType
ACCESS          read-write
STATUS          mandatory
DESCRIPTION
    "Type of diagnosis for base station.

    (From mibCtl ElementType 64 TypeOfBSDiagnosis)
    Description for mibCtl Type 251 DiagnosisType :
        Diagnosis Type.
    Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
::= { _625k-MCSysScalars 8 }

_625k-MCBSdiagnosisStatus              OBJECT-TYPE
SYNTAX          INTEGER -- DiagnosisStatusType

```

ACCESS read-only
STATUS mandatory
DESCRIPTION
"Diagnosis status for base station.

(From mibCtl ElementType 370 BSDiagnosisStatus)
Description for mibCtl Type 252 DiagnosisStatusType :
Diagnosis status Type.
Type derived from mibCtl Type 14 Unsigned32Type :
32 bit unsigned integer.
Type derived from mibCtl Type 11 Word32Type :
32 bits of raw opaque data.
Derived from basic 32 bit word type.
"
::= { _625k-MCSysScalars 9 }

_625k-MCBSdiagnosisFailReason OBJECT-TYPE
SYNTAX INTEGER -- DiagFailReasonType
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Diagnosis fail reason for base station.

(From mibCtl ElementType 371 BSDiagnosisFailReason)
Description for mibCtl Type 253 DiagFailReasonType :
Diagnosis fail reason Type.
Description for mibCtl DiagFailReasonType 1 PDSNPing :
Diagnosis fail reason is PDSN Ping.
Description for mibCtl DiagFailReasonType 2 Calibration :
Diagnosis fail reason is Calibration.
Description for mibCtl DiagFailReasonType 3 AntPath :
Diagnosis fail reason is TRx Antenna Path.
Description for mibCtl DiagFailReasonType 4 LOAlive :
Diagnosis fail reason is Local Oscillator DSP Alive.
Description for mibCtl DiagFailReasonType 5 GCLoopBack :
Diagnosis fail reason is GCLoopBack.
Description for mibCtl DiagFailReasonType 6 SlaveNM :
Diagnosis fail reason is Slave Modem control board.
Description for mibCtl DiagFailReasonType 7 GPSAnt :
Diagnosis fail reason is GPS Antenna.
Description for mibCtl DiagFailReasonType 8 SlotDSP :
Diagnosis fail reason is Modem control board DSP Alive.
Description for mibCtl DiagFailReasonType 9 ATMAlive :
Diagnosis fail reason is ATM Alive.
Description for mibCtl DiagFailReasonType 96 UndefinedName :
Diagnosis fail reason is Undefined Diag Name.
Description for mibCtl DiagFailReasonType 97 TimeOut :
Diagnosis fail reason is Time Out.
Description for mibCtl DiagFailReasonType 98 InvalidStateExec :
Diagnosis fail reason is Invalid State Execute.
Description for mibCtl DiagFailReasonType 99 ExecFail :
Diagnosis fail reason is Execute Fail.
"
::= { _625k-MCSysScalars 10 }

_625k-MCDiskDbUpdateSequence OBJECT-TYPE
SYNTAX Gauge -- Gauge32Type
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Base station Flash update sequence number.

This number is incremented on disk every time any other database element is actually changed on flash.
It is not incremented on redundant sets.
This number may also be set to a desired value.

(From mibCtl ElementType 6 DiskDbUpdateSequence)

Description for mibCtl Type 18 Gauge32Type :

32 bits of Gauge data.

Derived from basic 32 bit word type.

"

::= { _625k-MCSysScalars 11 }

_625k-MCStateOfBaseStation	OBJECT-TYPE
SYNTAX	INTEGER -- ComponentStateType
ACCESS	read-only
STATUS	mandatory
DESCRIPTION	

"State of base station as a whole.

This will not have values of Unknown or PowerOff since the base station would be unable to report such values.

When sufficiently initialized, the state will be Operating if accepting new calls (according to the desired state of the base station) or continuing ongoing calls;
or Ready if the desired state is Ready and there are no ongoing calls.

(From mibCtl ElementType 41 StateOfBaseStation)

Description for mibCtl Type 71 ComponentStateType :

Component operational state.

A component begins in the Unknown state.

If not detected, it enters and remains in the NotPresent state.

If detected, it enters the Uninitialized state, from where it may go to the Testing and Initializing states and then to the Standby or Operating state depending upon permissions.

Due to loss of permissions or resources, it may revert from the Operating state to the Standby state.

Due to failure or loss of permission, it may revert to the Uninitialized state, perhaps by way of the ShuttingDown state depending on the device.

From the Uninitialized state it may return to more advanced states depending upon permissions.

In case of a waiting period before (again) initializing, the component is considered to be Initializing.

Permissions include administrative permissions (from the operator); excessive failure restrictions; etc.

Description for mibCtl ComponentStateType 0 Unknown :

Component state not known.

Description for mibCtl ComponentStateType 1 NotPresent :

Component is not present.

Description for mibCtl ComponentStateType 2 PowerOff :

Component is present but powered off.

Description for mibCtl ComponentStateType 3 Uninitialized :

Component is present but not in use.

The power on/off state of the component is not specified in this case.

Description for mibCtl ComponentStateType 4 Testing :

Component is being tested.

Description for mibCtl ComponentStateType 5 Initializing :

Component is being initialized.

Description for mibCtl ComponentStateType 6 Ready :

Component is ready but not operating.
Description for mibCtl ComponentStateType 7 Operating :
Component is operating for normal use without restriction.

The component is either in actual use or may be used at any time,
without restriction.
Description for mibCtl ComponentStateType 8 Abandoned :
Component state is not the desired state due to excessive errors.

The component state is not that desired, and the Base Station
software has abandoned attempts to place the component in
the desired state.
The actual state of the component is undefined.
The Base Station software will resume attempting to place the
component in the desired state if the appropriate Reinitialize
action element is written with the correct value.
Also, the software may resume attempts under other conditions,
not all of which may be documented.
Description for mibCtl ComponentStateType 9 InitialSetUp :
Component is initial set up..

Initial set up state.
Description for mibCtl ComponentStateType 10 Degrading :
Component is degrading..

Degrading state.
Description for mibCtl ComponentStateType 11 Restriction :
Component is restriction..

Restriction state.
"
::= { _625k-MCSysScalars 12 }

_625k-MCBSTotalIndication OBJECT-TYPE
SYNTAX INTEGER -- IndicationType
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Status of BS total indicator.

In the current implementation,
this indication is set to the value On by BS.

(From mibCtl ElementType 525 BSTotalIndication)
Description for mibCtl Type 230 IndicationType :
Hardware indication status (LEDs).

Description for mibCtl IndicationType 0 Off :
Off.
Description for mibCtl IndicationType 1 Amber :
Amber.
Description for mibCtl IndicationType 2 Red :
Red.
Description for mibCtl IndicationType 3 Green :
Green.
Description for mibCtl IndicationType 4 NotPresent :
Not present.
"
::= { _625k-MCSysScalars 13 }

_625k-MCMasterAddress OBJECT-TYPE
SYNTAX INTEGER -- ModuleAddressType
ACCESS read-only


```

STATUS          mandatory
DESCRIPTION
    "Bus slot address of master Modem control board.

    This indicates which Modem control board is master
    of the base station.

    (From mibCtl ElementType 32 MasterAddress)
    Description for mibCtl Type 202 ModuleAddressType :
        Base station bus slot address.

        Most components of the Base Station for which data can
        be obtained are identified by a ModuleAddressType address
        and possibly a subsidiary address.
        [Limits: 0 7 ]
    Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
::= { _625k-MCSysScalars 14 }

_625k-MCBSManufactureID          OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Base Station manufacture identification number.

    The manufacture identification assigned by Vendor

    (From mibCtl ElementType 201 BSManufactureID)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
::= { _625k-MCSysScalars 15 }

_625k-MCBSSerialNumber          OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Base Station serial number.

    This is the character serial number of the base station.
    This serial number will be unique among all base stations
    of this type regardless of manufacturer.

    (From mibCtl ElementType 203 BSSerialNumber)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
::= { _625k-MCSysScalars 16 }

_625k-MCDiagnosisBaseStation          OBJECT-TYPE

```

```
SYNTAX          INTEGER -- BooleanType
ACCESS          read-write -- REALLY: write-only
STATUS          mandatory
DESCRIPTION
    "Diagnosis base station.

    This is a write-only element; only a value of TRUE is valid.

    (From mibCtl ElementType 47 DiagnosisBaseStation)
    Description for mibCtl Type 16 BooleanType :
        Truth value, 0=FALSE, 1=TRUE.

        This is a subset of TriStateType; no UNDEFINED value is provided.
        [Limits: 0 1 ]
    Description for mibCtl BooleanType 0 FALSE :
        False.
    Description for mibCtl BooleanType 1 TRUE :
        True.
    "
::= { _625k-MCSysScalars 17 }
```

```
_625k-MCRebootBaseStation          OBJECT-TYPE
SYNTAX          INTEGER -- BooleanType
ACCESS          read-write -- REALLY: write-only
STATUS          mandatory
DESCRIPTION
    "Reboot base station.

    This is a write-only element; only a value of TRUE is valid.
    All existing calls will be terminated abruptly.
    All components of the base station will be reinitialized
    according to the permanent contents of the Base Station database.
    The base station may be incommunicado for a period of time.

    The reinitialization may be delayed by a few seconds to allow
    for a clean shutdown.

    (From mibCtl ElementType 44 RebootBaseStation)
    Description for mibCtl Type 16 BooleanType :
        Truth value, 0=FALSE, 1=TRUE.

        This is a subset of TriStateType; no UNDEFINED value is provided.
        [Limits: 0 1 ]
    Description for mibCtl BooleanType 0 FALSE :
        False.
    Description for mibCtl BooleanType 1 TRUE :
        True.
    "
::= { _625k-MCSysScalars 18 }
```

```
_625k-MCBSModelNumber          OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Base Station model number.

    Base Station model number

    (From mibCtl ElementType 204 BSModelNumber)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
```

```

        Derived from basic 8 bit word type.
        "
        ::= { _625k-MCSysScalars 19 }

_625k-MCBSManufactureDate OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Base Station manufacture date.

        Base Station manufacture date

        (From mibCtl ElementType 205 BSManufactureDate)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
        ::= { _625k-MCSysScalars 20 }

_625k-MCBSHardwareRevision OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Base Station hardware revision.

        Base Station hardware revision

        (From mibCtl ElementType 206 BSHardwareRevision)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
        ::= { _625k-MCSysScalars 21 }

_625k-MCMiscComponents OBJECT IDENTIFIER
    -- DESCRIPTION      "Miscellaneous Component"
    ::= { IEEE802dot20-625k-MC-MIB 2 }

_625k-MCAntenna OBJECT IDENTIFIER
    -- DESCRIPTION      "Antenna"
    ::= { _625k-MCMiscComponents 1 }

_625k-MCAntennaTable OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCAntennaTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION      "Antenna Table"
    ::= { _625k-MCAntenna 1 }

_625k-MCAntennaTableEntry OBJECT-TYPE
    SYNTAX      _625k-MCAntennaTableEntry

```

```

ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     ""
INDEX { _625k-MCAntennaTableIndex }
 ::= { _625k-MCAntennaTable 1 }

_625k-MCAntennaTableEntry ::= SEQUENCE {
    _625k-MCAntennaTableIndex    INTEGER, -- AntennaAddressType
    _625k-MCStateOfAntenna      INTEGER -- ComponentStateType
}

_625k-MCAntennaTableIndex OBJECT-TYPE
SYNTAX      INTEGER -- AntennaAddressType
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "
    Description for mibCtl Type 210 AntennaAddressType :
        Component antenna address.
        [Limits: 0 11 ]
    Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
"
 ::= { _625k-MCAntennaTableEntry 1 }

_625k-MCStateOfAntenna OBJECT-TYPE
SYNTAX      INTEGER -- ComponentStateType
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "
    State of Antenna as a whole.

    (From mibCtl ElementType 211 StateOfAntenna)
    Description for mibCtl Type 71 ComponentStateType :
        Component operational state.

        A component begins in the Unknown state.
        If not detected, it enters and remains in the NotPresent state.
        If detected, it enters the Uninitialized state, from where it
        may go to the Testing and Initializing states and then to the
        Standby or Operating state depending upon permissions.
        Due to loss of permissions or resources, it may revert from
        the Operating state to the Standby state.
        Due to failure or loss of permission, it may revert to the
        Uninitialized state, perhaps by way of the ShuttingDown state
        depending on the device.
        From the Uninitialized state it may return to more advanced
        states depending upon permissions.
        In case of a waiting period before (again) initializing,
        the component is considered to be Initializing.

        Permissions include administrative permissions (from the
        operator); excessive failure restrictions; etc.
    Description for mibCtl ComponentStateType 0 Unknown :
        Component state not known.
    Description for mibCtl ComponentStateType 1 NotPresent :
        Component is not present.
    Description for mibCtl ComponentStateType 2 PowerOff :
        Component is present but powered off.
    Description for mibCtl ComponentStateType 3 Uninitialized :
        Component is present but not in use.

```

The power on/off state of the component is not specified in this case.

Description for mibCtl ComponentStateType 4 Testing :
Component is being tested.

Description for mibCtl ComponentStateType 5 Initializing :
Component is being initialized.

Description for mibCtl ComponentStateType 6 Ready :
Component is ready but not operating.

Description for mibCtl ComponentStateType 7 Operating :
Component is operating for normal use without restriction.

The component is either in actual use or may be used at any time, without restriction.

Description for mibCtl ComponentStateType 8 Abandoned :
Component state is not the desired state due to excessive errors.

The component state is not that desired, and the Base Station software has abandoned attempts to place the component in the desired state.
The actual state of the component is undefined.
The Base Station software will resume attempting to place the component in the desired state if the appropriate Reinitialize action element is written with the correct value.
Also, the software may resume attempts under other conditions, not all of which may be documented.

Description for mibCtl ComponentStateType 9 InitialSetUp :
Component is initial set up..

Initial set up state.

Description for mibCtl ComponentStateType 10 Degrading :
Component is degrading..

Degrading state.

Description for mibCtl ComponentStateType 11 Restriction :
Component is restriction..

Restriction state.

"

```
 ::= { _625k-MCAntennaTableEntry 2 }
```

_625k-MCBSTemperatures OBJECT IDENTIFIER

```
-- DESCRIPTION "BS Temperature"
 ::= { _625k-MCMiscComponents 3 }
```

_625k-MCBSTemperature OBJECT-TYPE

```
SYNTAX OCTET STRING (SIZE(0..4)) -- DegreesCelsiusType
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The temperature of Base station (degrees Celsius).

    The latest recorded temperature of a given BS.

    (From mibCtl ElementType 536 BSTemperature)
    Description for mibCtl Type 807 DegreesCelsiusType :
        Temperature in degrees Celsius.
    Type derived from mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
 ::= { _625k-MCBSTemperatures 1 }
```

```

_625k-MCCableInfo                                OBJECT IDENTIFIER
-- DESCRIPTION          "Cable Info"
 ::= { _625k-MCMiscComponents 4 }

_625k-MCCableLossValueForLoCal                    OBJECT-TYPE
SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "Value of cable loss for Local Oscillator.

Value of cable loss.

(From mibCtl ElementType 801 CableLossValueForLoCal)
Description for mibCtl Type 15 TextType :
    ASCII or compatible text.
Type derived from mibCtl Type 12 OctetType :
    8 bits of raw opaque data.
Derived from basic 8 bit word type.
"
 ::= { _625k-MCCableInfo 1 }

_625k-MCAntCableTable                            OBJECT-TYPE
SYNTAX SEQUENCE OF _625k-MCAntCableTableEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     "Antenna Cable"
 ::= { _625k-MCCableInfo 2 }

_625k-MCAntCableTableEntry                        OBJECT-TYPE
SYNTAX          _625k-MCAntCableTableEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     ""
INDEX { _625k-MCAntCableTableIndex }
 ::= { _625k-MCAntCableTable 1 }

_625k-MCAntCableTableEntry ::= SEQUENCE {
    _625k-MCAntCableTableIndex      INTEGER, -- AntennaAddressType
    _625k-MCCableLossValueForAntenna OCTET STRING (SIZE(0..20)) --
    TextType X 20
}

_625k-MCAntCableTableIndex                        OBJECT-TYPE
SYNTAX          INTEGER -- AntennaAddressType
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "
    Description for mibCtl Type 210 AntennaAddressType :
        Component antenna address.
        [Limits: 0 11 ]
    Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.

```

```

"
 ::= { _625k-MCAntCableTableEntry 1 }

_625k-MCCableLossValueForAntenna OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Value of cable loss for Antenna.

        Value of cable loss.

        (From mibCtl ElementType 802 CableLossValueForAntenna)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
"
 ::= { _625k-MCAntCableTableEntry 2 }

_625k-MCGPSCableTable OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCGPSCableTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "GPS Cable"
    ::= { _625k-MCCableInfo 10 }

_625k-MCGPSCableTableEntry OBJECT-TYPE
    SYNTAX      _625k-MCGPSCableTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION ""
    INDEX { _625k-MCGPSCableTableIndex }
    ::= { _625k-MCGPSCableTable 1 }

_625k-MCGPSCableTableEntry ::= SEQUENCE {
    _625k-MCGPSCableTableIndex INTEGER, -- GpsAddressType
    _625k-MCCableLengthForGps OCTET STRING (SIZE(0..20)) --
    TextType X 20
}

_625k-MCGPSCableTableIndex OBJECT-TYPE
    SYNTAX      INTEGER -- GpsAddressType
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "
        Description for mibCtl Type 209 GpsAddressType :
            Base station GPS component address.

            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
"
    ::= { _625k-MCGPSCableTableEntry 1 }

```

```

_625k-MCCableLengthForGps                                OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION
        "Cable length for Gps.

        This cable is used for calibration.

        (From mibCtl ElementType 803 CableLengthForGps)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { _625k-MCGPSCableTableEntry 2 }

_625k-MCGPS                                               OBJECT IDENTIFIER
    -- DESCRIPTION          "GPS"
    ::= { _625k-MCMiscComponents 6 }

_625k-MCGPSTable                                          OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCGPSTableEntry
    ACCESS           not-accessible
    STATUS           mandatory
    DESCRIPTION      "GPS Table"
    ::= { _625k-MCGPS 1 }

_625k-MCGPSTableEntry                                    OBJECT-TYPE
    SYNTAX          _625k-MCGPSTableEntry
    ACCESS           not-accessible
    STATUS           mandatory
    DESCRIPTION      ""
    INDEX { _625k-MCGPSTableIndex }
    ::= { _625k-MCGPSTable 1 }

_625k-MCGPSTableEntry ::= SEQUENCE {
    _625k-MCGPSTableIndex          INTEGER, -- GpsAddressType
    _625k-MCStateOfGps             INTEGER, -- ComponentStateType
    _625k-MCGpsNumberOfSatelliteSeen INTEGER, -- Unsigned32Type
    _625k-MCGpsIndication           INTEGER, -- IndicationType
    _625k-MCGpsSerialNumber         OCTET STRING (SIZE(0..20)) --
    TextType X 20
}

_625k-MCGPSTableIndex                                    OBJECT-TYPE
    SYNTAX          INTEGER -- GpsAddressType
    ACCESS           read-only
    STATUS           mandatory
    DESCRIPTION      "
        Description for mibCtl Type 209 GpsAddressType :
            Base station GPS component address.

        [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :

```



```

    32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
    32 bits of raw opaque data.
    Derived from basic 32 bit word type.
"
 ::= { _625k-MCGPSTableEntry 1 }

_625k-MCStateOfGps                                OBJECT-TYPE
    SYNTAX      INTEGER -- ComponentStateType
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "GPS state.

        The state of the GPS (Global Positioning System)
        on the active local oscillator unit

        (From mibCtl ElementType 420 StateOfGps)
        Description for mibCtl Type 71 ComponentStateType :
        Component operational state.

        A component begins in the Unknown state.
        If not detected, it enters and remains in the NotPresent state.
        If detected, it enters the Uninitialized state, from where it
        may go to the Testing and Initializing states and then to the
        Standby or Operating state depending upon permissions.
        Due to loss of permissions or resources, it may revert from
        the Operating state to the Standby state.
        Due to failure or loss of permission, it may revert to the
        Uninitialized state, perhaps by way of the ShuttingDown state
        depending on the device.
        From the Uninitialized state it may return to more advanced
        states depending upon permissions.
        In case of a waiting period before (again) initializing,
        the component is considered to be Initializing.

        Permissions include administrative permissions (from the
        operator); excessive failure restrictions; etc.
        Description for mibCtl ComponentStateType 0 Unknown :
        Component state not known.
        Description for mibCtl ComponentStateType 1 NotPresent :
        Component is not present.
        Description for mibCtl ComponentStateType 2 PowerOff :
        Component is present but powered off.
        Description for mibCtl ComponentStateType 3 Uninitialized :
        Component is present but not in use.

        The power on/off state of the component is not specified in
        this case.
        Description for mibCtl ComponentStateType 4 Testing :
        Component is being tested.
        Description for mibCtl ComponentStateType 5 Initializing :
        Component is being initialized.
        Description for mibCtl ComponentStateType 6 Ready :
        Component is ready but not operating.
        Description for mibCtl ComponentStateType 7 Operating :
        Component is operating for normal use without restriction.

        The component is either in actual use or may be used at any time,
        without restriction.
        Description for mibCtl ComponentStateType 8 Abandoned :
        Component state is not the desired state due to excessive errors.

        The component state is not that desired, and the Base Station
        software has abandoned attempts to place the component in
        the desired state.
        The actual state of the component is undefined.

```

The Base Station software will resume attempting to place the component in the desired state if the appropriate Reinitialize action element is written with the correct value.
Also, the software may resume attempts under other conditions, not all of which may be documented.

Description for mibCtl ComponentStateType 9 InitialSetUp :
Component is initial set up..

Initial set up state.

Description for mibCtl ComponentStateType 10 Degrading :
Component is degrading..

Degrading state.

Description for mibCtl ComponentStateType 11 Restriction :
Component is restriction..

Restriction state.

"

```
::= { _625k-MCGPSTableEntry 2 }
```

```
_625k-MCGpsNumberOfSatelliteSeen          OBJECT-TYPE
SYNTAX          INTEGER -- Unsigned32Type
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Number of satellites seen by GPS.

    The number of satellites seen by the GPS (Global Positioning System)
    on the active local oscillator unit

    (From mibCtl ElementType 421 GpsNumberOfSatelliteSeen)
    Description for mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
    "
::= { _625k-MCGPSTableEntry 3 }
```

```
_625k-MCGpsIndication                      OBJECT-TYPE
SYNTAX          INTEGER -- IndicationType
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Status of GPS indicator.

    (From mibCtl ElementType 530 GpsIndication)
    Description for mibCtl Type 230 IndicationType :
        Hardware indication status (LEDs).

    Description for mibCtl IndicationType 0 Off :
        Off.
    Description for mibCtl IndicationType 1 Amber :
        Amber.
    Description for mibCtl IndicationType 2 Red :
        Red.
    Description for mibCtl IndicationType 3 Green :
        Green.
    Description for mibCtl IndicationType 4 NotPresent :
        Not present.
    "
```

```

 ::= { _625k-MCGPSTableEntry 4 }

_625k-MCGpsSerialNumber OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "GPS serial number text.

        Factory set uniquely for each component.

        (From mibCtl ElementType 570 GpsSerialNumber)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
 ::= { _625k-MCGPSTableEntry 5 }

_625k-MCPowerAmplifier OBJECT IDENTIFIER
    -- DESCRIPTION      "Power Amplifier"
 ::= { _625k-MCMiscComponents 8 }

_625k-MCPAUnitTable OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCPAUnitTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION      "PA Table"
 ::= { _625k-MCPowerAmplifier 1 }

_625k-MCPAUnitTableEntry OBJECT-TYPE
    SYNTAX      _625k-MCPAUnitTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION      ""
    INDEX { _625k-MCPAUnitTableIndex }
 ::= { _625k-MCPAUnitTable 1 }

_625k-MCPAUnitTableEntry ::= SEQUENCE {
    _625k-MCPAUnitTableIndex      INTEGER, -- PAUnitAddressType
    _625k-MCRebootPAUnit          INTEGER, -- BooleanType
    _625k-MCStateOfPAUnit         INTEGER, -- ComponentStateType
    _625k-MCPAUnitIndication      INTEGER, -- IndicationType
    _625k-MCPAUnitSerialNumber    OCTET STRING (SIZE(0..20)), --
    TextType X 20
    _625k-MCPAUnitModelNumber     OCTET STRING (SIZE(0..20)), --
    TextType X 20
    _625k-MCPAUnitManufactureDate OCTET STRING (SIZE(0..20)), --
    TextType X 20
    _625k-MCPAUnitHardwareRevision OCTET STRING (SIZE(0..20)), --
    TextType X 20
    _625k-MCPAUnitManufactureID   OCTET STRING (SIZE(0..20)), --
    TextType X 20
    _625k-MCPAUnitTemperature     OCTET STRING (SIZE(0..4)) --
    DegreesCelsiusType
}

```

```

_625k-MCPAUnitTableIndex                                OBJECT-TYPE
SYNTAX          INTEGER -- PAUnitAddressType
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "
    Description for mibCtl Type 207 PAUnitAddressType :
        Base station power amplifier component unit address.

    A power amplifier unit address is a subset of Base Station
    component addresses,
    restricted to power amplifier components only.
    Power amplifiers boost radio frequency
    signal levels.
    [Limits: 0 3 ]
    Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
    Derived from basic 32 bit word type.
"
::= { _625k-MCPAUnitTableEntry 1 }

_625k-MCRebootPAUnit                                    OBJECT-TYPE
SYNTAX          INTEGER -- BooleanType
ACCESS          read-write -- REALLY: write-only
STATUS          mandatory
DESCRIPTION     "Action to reboot a PA unit.

    This is a write-only element; only a value of TRUE is valid.

    (From mibCtl ElementType 506 RebootPAUnit)
    Description for mibCtl Type 16 BooleanType :
        Truth value, 0=FALSE, 1=TRUE.

    This is a subset of TriStateType; no UNDEFINED value is provided.
    [Limits: 0 1 ]
    Description for mibCtl BooleanType 0 FALSE :
        False.
    Description for mibCtl BooleanType 1 TRUE :
        True.
"
::= { _625k-MCPAUnitTableEntry 2 }

_625k-MCStateOfPAUnit                                    OBJECT-TYPE
SYNTAX          INTEGER -- ComponentStateType
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "State of PA as a whole.

    (From mibCtl ElementType 212 StateOfPAUnit)
    Description for mibCtl Type 71 ComponentStateType :
        Component operational state.

    A component begins in the Unknown state.
    If not detected, it enters and remains in the NotPresent state.
    If detected, it enters the Uninitialized state, from where it
    may go to the Testing and Initializing states and then to the
    Standby or Operating state depending upon permissions.
    Due to loss of permissions or resources, it may revert from
    the Operating state to the Standby state.

```

Due to failure or loss of permission, it may revert to the Uninitialized state, perhaps by way of the ShuttingDown state depending on the device.

From the Uninitialized state it may return to more advanced states depending upon permissions.

In case of a waiting period before (again) initializing, the component is considered to be Initializing.

Permissions include administrative permissions (from the operator); excessive failure restrictions; etc.

Description for mibCtl ComponentStateType 0 Unknown :

Component state not known.

Description for mibCtl ComponentStateType 1 NotPresent :

Component is not present.

Description for mibCtl ComponentStateType 2 PowerOff :

Component is present but powered off.

Description for mibCtl ComponentStateType 3 Uninitialized :

Component is present but not in use.

The power on/off state of the component is not specified in this case.

Description for mibCtl ComponentStateType 4 Testing :

Component is being tested.

Description for mibCtl ComponentStateType 5 Initializing :

Component is being initialized.

Description for mibCtl ComponentStateType 6 Ready :

Component is ready but not operating.

Description for mibCtl ComponentStateType 7 Operating :

Component is operating for normal use without restriction.

The component is either in actual use or may be used at any time, without restriction.

Description for mibCtl ComponentStateType 8 Abandoned :

Component state is not the desired state due to excessive errors.

The component state is not that desired, and the Base Station software has abandoned attempts to place the component in the desired state.

The actual state of the component is undefined.

The Base Station software will resume attempting to place the component in the desired state if the appropriate Reinitialize action element is written with the correct value.

Also, the software may resume attempts under other conditions, not all of which may be documented.

Description for mibCtl ComponentStateType 9 InitialSetUp :

Component is initial set up..

Initial set up state.

Description for mibCtl ComponentStateType 10 Degrading :

Component is degrading..

Degrading state.

Description for mibCtl ComponentStateType 11 Restriction :

Component is restriction..

Restriction state.

"

::= { _625k-MCPAUnitTableEntry 3 }

_625k-MCPAUnitIndication	OBJECT-TYPE
SYNTAX	INTEGER -- IndicationType
ACCESS	read-only
STATUS	mandatory
DESCRIPTION	"Status of PA Unit indicator.

(From mibCtl ElementType 526 PAUnitIndication)
Description for mibCtl Type 230 IndicationType :
Hardware indication status (LEDs).

Description for mibCtl IndicationType 0 Off :
Off.
Description for mibCtl IndicationType 1 Amber :
Amber.
Description for mibCtl IndicationType 2 Red :
Red.
Description for mibCtl IndicationType 3 Green :
Green.
Description for mibCtl IndicationType 4 NotPresent :
Not present.

"
::= { _625k-MCPAUnitTableEntry 4 }

_625k-MCPAUnitSerialNumber OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS read-only
STATUS mandatory
DESCRIPTION
"PA unit serial number text.

Factory set uniquely for each component.

(From mibCtl ElementType 560 PAUnitSerialNumber)
Description for mibCtl Type 15 TextType :
ASCII or compatible text.
Type derived from mibCtl Type 12 OctetType :
8 bits of raw opaque data.
Derived from basic 8 bit word type.

"
::= { _625k-MCPAUnitTableEntry 5 }

_625k-MCPAUnitModelNumber OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS read-only
STATUS mandatory
DESCRIPTION
"PA unit model number.

Factory set with description of component type, including
the major revision level.

(From mibCtl ElementType 561 PAUnitModelNumber)
Description for mibCtl Type 15 TextType :
ASCII or compatible text.
Type derived from mibCtl Type 12 OctetType :
8 bits of raw opaque data.
Derived from basic 8 bit word type.

"
::= { _625k-MCPAUnitTableEntry 6 }

_625k-MCPAUnitManufactureDate OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS read-only
STATUS mandatory
DESCRIPTION
"PA unit manufacture date.

```

    Factory set to month and date of manufacture date of the module.

    (From mibCtl ElementType 562 PAUnitManufactureDate)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
 ::= { _625k-MCPAUnitTableEntry 7 }

_625k-MCPAUnitHardwareRevision          OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "PA unit hardware revision name.

    Set at the factory to indicate the minor hardware revision
    level of the module.

    (From mibCtl ElementType 563 PAUnitHardwareRevision)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
 ::= { _625k-MCPAUnitTableEntry 8 }

_625k-MCPAUnitManufactureID             OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..20)) -- TextType X 20
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "PA unit manufacture ID.

    (From mibCtl ElementType 564 PAUnitManufactureID)
    Description for mibCtl Type 15 TextType :
        ASCII or compatible text.
    Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
    Derived from basic 8 bit word type.
    "
 ::= { _625k-MCPAUnitTableEntry 9 }

_625k-MCPAUnitTemperature                OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(0..4)) -- DegreesCelsiusType
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "The temperature of PAUnit (degrees Celsius).

    The latest recorded temperature of a given PAUnit.

    (From mibCtl ElementType 539 PAUnitTemperature)
    Description for mibCtl Type 807 DegreesCelsiusType :
        Temperature in degrees Celsius.
    Type derived from mibCtl Type 15 TextType :

```

```

        ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
        8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
 ::= { _625k-MCPAUnitTableEntry 10 }

```

```

_625k-MCPAModuleTable OBJECT-TYPE
    SYNTAX SEQUENCE OF _625k-MCPAModuleTableEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION "PA Module Table"
    ::= { _625k-MCPowerAmplifier 2 }

```

```

_625k-MCPAModuleTableEntry OBJECT-TYPE
    SYNTAX _625k-MCPAModuleTableEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION ""
    INDEX { _625k-MCPAModuleTableIndex1, _625k-MCPAModuleTableIndex2 }
    ::= { _625k-MCPAModuleTable 1 }

```

```

_625k-MCPAModuleTableEntry ::= SEQUENCE {
    _625k-MCPAModuleTableIndex1 INTEGER, -- PAUnitAddressType
    _625k-MCPAModuleTableIndex2 INTEGER -- PAModuleAddressType
}

```

```

_625k-MCPAModuleTableIndex1 OBJECT-TYPE
    SYNTAX INTEGER -- PAUnitAddressType
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "
        Description for mibCtl Type 207 PAUnitAddressType :
        Base station power amplifier component unit address.

        A power amplifier unit address is a subset of Base Station
        component addresses,
        restricted to power amplifier components only.
        Power amplifiers boost radio frequency
        signal levels.
        [Limits: 0 3 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { _625k-MCPAModuleTableEntry 1 }

```

```

_625k-MCPAModuleTableIndex2 OBJECT-TYPE
    SYNTAX INTEGER -- PAModuleAddressType
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "
        Description for mibCtl Type 208 PAModuleAddressType :
        Base station power amplifier component module address.

        A power amplifier module address is a subset of Base Station
        component addresses,

```



```

        [Limits: 0 2 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
        32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
        32 bits of raw opaque data.
        Derived from basic 32 bit word type.
"
 ::= { _625k-MCPAModuleTableEntry 2 }

_625k-MCPowerSupply                                OBJECT IDENTIFIER
-- DESCRIPTION "Power supply"
 ::= { _625k-MCMiscComponents 9 }

_625k-MCPowerSupplyTable                            OBJECT-TYPE
SYNTAX SEQUENCE OF _625k-MCPowerSupplyTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Power supply Table"
 ::= { _625k-MCPowerSupply 1 }

_625k-MCPowerSupplyTableEntry                        OBJECT-TYPE
SYNTAX _625k-MCPowerSupplyTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION ""
INDEX { _625k-MCPowerSupplyTableIndex }
 ::= { _625k-MCPowerSupplyTable 1 }

_625k-MCPowerSupplyTableEntry ::= SEQUENCE {
    _625k-MCPowerSupplyTableIndex INTEGER, -- PowerAddressType
    _625k-MCStateOfPowerSupply    INTEGER, -- ComponentStateType
    _625k-MCPowerSupplyIndication INTEGER -- IndicationType
}

_625k-MCPowerSupplyTableIndex                        OBJECT-TYPE
SYNTAX INTEGER -- PowerAddressType
ACCESS read-only
STATUS mandatory
DESCRIPTION "
    Description for mibCtl Type 211 PowerAddressType :
    Component power supply address.
    [Limits: 0 2 ]
    Type derived from mibCtl Type 14 Unsigned32Type :
    32 bit unsigned integer.
    Type derived from mibCtl Type 11 Word32Type :
    32 bits of raw opaque data.
    Derived from basic 32 bit word type.
"
 ::= { _625k-MCPowerSupplyTableEntry 1 }

_625k-MCStateOfPowerSupply                            OBJECT-TYPE
SYNTAX INTEGER -- ComponentStateType
ACCESS read-only
STATUS mandatory
DESCRIPTION "State of Power supply as a whole.

```

(From mibCtl ElementType 213 StateOfPowerSupply)
Description for mibCtl Type 71 ComponentStateType :
Component operational state.

A component begins in the Unknown state.
If not detected, it enters and remains in the NotPresent state.
If detected, it enters the Uninitialized state, from where it may go to the Testing and Initializing states and then to the Standby or Operating state depending upon permissions.
Due to loss of permissions or resources, it may revert from the Operating state to the Standby state.
Due to failure or loss of permission, it may revert to the Uninitialized state, perhaps by way of the ShuttingDown state depending on the device.
From the Uninitialized state it may return to more advanced states depending upon permissions.
In case of a waiting period before (again) initializing, the component is considered to be Initializing.

Permissions include administrative permissions (from the operator); excessive failure restrictions; etc.
Description for mibCtl ComponentStateType 0 Unknown :
Component state not known.
Description for mibCtl ComponentStateType 1 NotPresent :
Component is not present.
Description for mibCtl ComponentStateType 2 PowerOff :
Component is present but powered off.
Description for mibCtl ComponentStateType 3 Uninitialized :
Component is present but not in use.

The power on/off state of the component is not specified in this case.
Description for mibCtl ComponentStateType 4 Testing :
Component is being tested.
Description for mibCtl ComponentStateType 5 Initializing :
Component is being initialized.
Description for mibCtl ComponentStateType 6 Ready :
Component is ready but not operating.
Description for mibCtl ComponentStateType 7 Operating :
Component is operating for normal use without restriction.

The component is either in actual use or may be used at any time, without restriction.
Description for mibCtl ComponentStateType 8 Abandoned :
Component state is not the desired state due to excessive errors.

The component state is not that desired, and the Base Station software has abandoned attempts to place the component in the desired state.
The actual state of the component is undefined.
The Base Station software will resume attempting to place the component in the desired state if the appropriate Reinitialize action element is written with the correct value.
Also, the software may resume attempts under other conditions, not all of which may be documented.
Description for mibCtl ComponentStateType 9 InitialSetUp :
Component is initial set up..

Initial set up state.
Description for mibCtl ComponentStateType 10 Degrading :
Component is degrading..

Degrading state.
Description for mibCtl ComponentStateType 11 Restriction :
Component is restriction..

Restriction state.

```

"
 ::= { _625k-MCPowerSupplyTableEntry 2 }

_625k-MCPowerSupplyIndication          OBJECT-TYPE
SYNTAX          INTEGER -- IndicationType
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
    "Status of Power Supply indicator.

    (From mibCtl ElementType 527 PowerSupplyIndication)
    Description for mibCtl Type 230 IndicationType :
        Hardware indication status (LEDs).

    Description for mibCtl IndicationType 0 Off :
        Off.
    Description for mibCtl IndicationType 1 Amber :
        Amber.
    Description for mibCtl IndicationType 2 Red :
        Red.
    Description for mibCtl IndicationType 3 Green :
        Green.
    Description for mibCtl IndicationType 4 NotPresent :
        Not present.
"
 ::= { _625k-MCPowerSupplyTableEntry 3 }
END

```

Annex A

(informative)

625K-MC Glossary of Technical Terms

As specified in the Annex A of the baseline specification ATIS-0700004.2005, which contains a glossary of technical terms used in the *HC-SDMA* Protocol Specification.

Annex B

(informative)

Bibliography

- [B1] IEEE 100™, *The Authoritative Dictionary of IEEE Standards Terms*, 7th Edition.
- [B2] IETF RFC 2212, Specification of Guaranteed Quality of Service.
- [B3] IETF RFC 2215, General Characterization Parameters for Integrated Service Network Elements.
- [B4] TIA-1121.000, Overview for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B5] TIA-1121.001, Physical Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B6] TIA-1121.002, MAC Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B7] TIA-1121.003, Radio Link Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B8] TIA-1121.004, Application Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B9] TIA-1121.005, Security Functions for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B10] TIA-1121.006, Connection Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B11] TIA-1121.007, Session Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B12] TIA-1121.008, Route Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification.
- [B13] TIA-1121.009, Broadcast-Multicast Upper Layers for Ultra Mobile Broadband (UMB) Air Interface Specification.