IEEE Std 802.16-2001™

IEEE Standards

802.16™

IEEE Standard for
Local and metropolitan area networks

Part 16: Air Interface for Fixed Broadband Wireless Access Systems

IEEE Computer Society
and the
IEEE Microwave Theory and Techniques Society

Sponsored by the
LAN/MAN Standards Committee

IEEE

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

8 April 2002

Print: SH94964
PDF: SS94964

IEEE Std 802.16™-2001

IEEE Standard for
Local and metropolitan area networks

Part 16: Air Interface for Fixed Broadband Wireless Access Systems

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

and the
IEEE Microwave Theory and Techniques Society

Approved 6 December 2001

IEEE-SA Standards Board

Abstract: This standard specifies the air interface of fixed (stationary) point-to-multipoint broadband wireless access systems providing multiple services. The medium access control layer is capable of supporting multiple physical layer specifications optimized for the frequency bands of application. The standard includes a particular physical layer specification applicable to systems operating between 10 and 66 GHz.

Keywords: fixed broadband wireless access network, metropolitan area network, microwave, millimeter wave, WirelessMAN™ standards

Grateful acknowledgment is made to Cable Television Laboratories for the permission to use the following source material:

Radio Frequency Interface Specification (version 1.1), part of Data-Over-Cable Service Interface Specifications, © Copyright 1999, Cable Television Laboratories.

Baseline Privacy Plus Interface Specification, © Copyright 1999, Cable Television Laboratories.

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

Print: ISBN 0-7381-3070-2 SH94964
PDF: ISBN 0-7381-3071-0 SS94964

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 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 the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

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.

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 addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: 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 shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

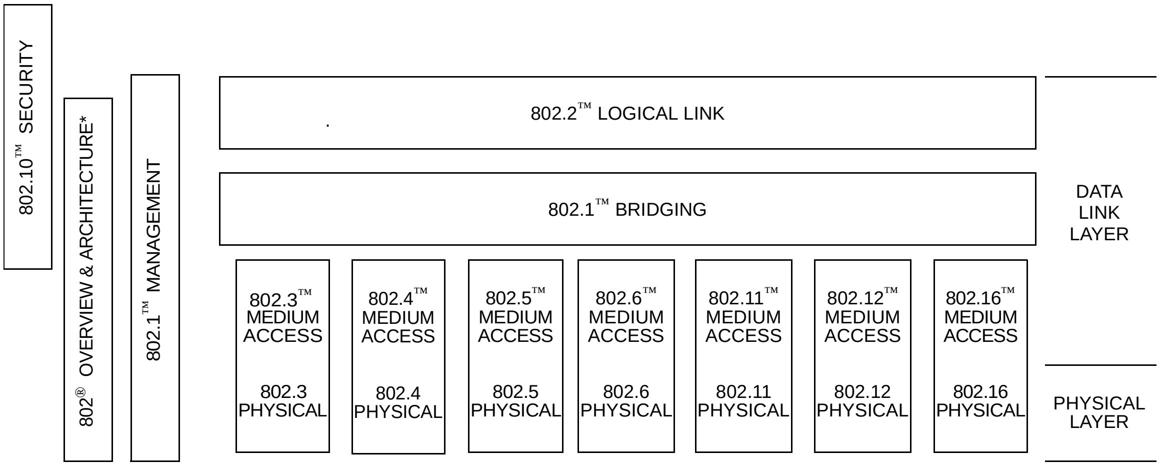
Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright 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 802.16-2001, IEEE Standard for Local and metropolitan area networks—Part 16: Air Interface for Fixed Broadband Wireless Access Systems.)

This standard specifies the air interface of a fixed (stationary) point-to-multipoint broadband wireless access system providing multiple services in a wireless metropolitan area network (MAN). The WirelessMAN™ medium access control layer defined here is capable of supporting multiple physical layer specifications optimized for the frequency bands of application. The standard includes a particular physical layer specification applicable to systems operating between 10 and 66 GHz. This 10–66 GHz air interface, based on single-carrier modulation, is known as the WirelessMAN-SC™ air interface. An amendment to this standard, to support 2–11 GHz using an enhanced version of the same basic medium access control layer along with new physical layer specifications, is in development in IEEE-SA Project 802.16a.

This standard is part of a family of standards for local and metropolitan area networks. The relationship between the standard and other members of the family is shown below. (The numbers in the figure refer to IEEE standards numbers.[1])



* Formerly IEEE Std 802.1A™.

This family of standards deals with the Physical and Data Link Layers as defined by the International Organization for Standardization (ISO) Open Systems Interconnection Basic Reference Model (ISO/IEC 7498-1:1994). The access standards define several types of medium access technologies and associated physical media, each appropriate for particular applications or system objectives. Other types are under investigation.

The standards defining the technologies noted above are as follows:

- IEEE Std 802:[2] *Overview and Architecture*. This standard provides an overview to the family of IEEE 802 Standards. This document forms part of the IEEE Std 802.1 scope of work.

[1] The IEEE standards referred to in the above figure and list are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

[2] The IEEE 802 Architecture and Overview Specification, originally known as IEEE Std 802.1A, has been renumbered as IEEE Std 802. This has been done to accommodate recognition of the base standard in a family of standards. References to IEEE Std 802.1A should be considered as references to IEEE Std 802.

- IEEE Std 802.1B™ and 802.1K™ [ISO/IEC 15802-2]: *LAN/MAN Management*. Defines an Open Systems Interconnection (OSI) management-compatible architecture, and services and protocol elements for use in a LAN/MAN environment for performing remote management.

- IEEE Std 802.1D™ *Media Access Control* (*MAC) Bridges*. Specifies an architecture and protocol for the [ISO/IEC 15802-3]: interconnection of IEEE 802 LANs below the MAC service boundary.

- IEEE Std 802.1E™ [ISO/IEC 15802-4]: *System Load Protocol*. Specifies a set of services and protocol for those aspects of management concerned with the loading of systems on IEEE 802 LANs.

- IEEE Std 802.1F™ *Common Definitions and Procedures for IEEE 802 Management Information*.

- IEEE Std 802.1G™ [ISO/IEC 15802-5]: *Remote Media Access Control (MAC) Bridging*. Specifies extensions for the interconnection, using non-LAN systems communication technologies, of geographically separated IEEE 802 LANs below the level of the logical link control protocol.

- IEEE Std 802.1H™ [ISO/IEC TR 11802-5] *Recommended Practice for Media Access Control (MAC) Bridging of Ethernet V2.0 in IEEE 802 Local Area Networks*.

- IEEE Std 802.1Q™ *Virtual Bridged Local Area Networks*. Defines an architecture for Virtual Bridged LANs, the services provided in Virtual Bridged LANs, and the protocols and algorithms involved in the provision of those services.

- IEEE Std 802.2 [ISO/IEC 8802-2]: *Logical Link Control*.

- IEEE Std 802.3 [ISO/IEC 8802-3]: *CSMA/CD Access Method and Physical Layer Specifications*.

- IEEE Std 802.4 [ISO/IEC 8802-4]: *Token Bus Access Method and Physical Layer Specifications*.

- IEEE Std 802.5 [ISO/IEC 8802-5]: *Token Ring Access Method and Physical Layer Specifications*.

- IEEE Std 802.6 [ISO/IEC 8802-6]: *Distributed Queue Dual Bus Access Method and Physical Layer Specifications*.

- IEEE Std 802.10: *Interoperable LAN/MAN Security*. Currently approved: Secure Data Exchange (SDE).

- IEEE Std 802.11: [ISO/IEC 8802-11] *Wireless LAN Medium Access Control (MAC) Sublayer and Physical Layer Specifications.*

- IEEE Std 802.12: [ISO/IEC 8802-12] *Demand Priority Access Method, Physical Layer and Repeater Specification*.

- IEEE Std 802.15: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for: Wireless Personal Area Networks.*

- IEEE Std 802.16: *Air Interface for Fixed Broadband Wireless Access Systems*.
- IEEE Std 802.17™: *Resilient Packet Ring Access Method and Physical Layer Specifications.*

In addition to the family of standards, the following is a recommended practice for a common physical layer technology:

- IEEE Std 802.7™: *IEEE Recommended Practice for Broadband Local Area Networks*.

The reader of this standard is urged to become familiar with the complete family of standards.

Conformance test methodology

An additional standards series, identified by the number 1802™, has been established to identify the conformance test methodology documents for the IEEE 802 family of standards. For example, the conformance test documents for IEEE 802.3 are numbered 1802.3™.

Interpretations and errata

Interpretations and errata associated with this standard may be found at one of the following Internet locations:

— http://standards.ieee.org/reading/ieee/interp/

— http://standards.ieee.org/reading/ieee/updates/errata/

One such interpretation exists at the time of publication.

Participants

This document was developed by the IEEE 802.16 Working Group on Broadband Wireless Access, which is responsible for Wireless Metropolitan Area Network (WirelessMAN™) Standards. The 802.16 Working Group had the following officers:

Roger B. Marks, *Chair*
Brian G. Kiernan, *Vice Chair*
Carl J. Bushue, *Secretary*

Louis Olsen served as Working Group Vice Chair during the initial development of this document, until September 2000. J. Scott Marin served as Secretary during the draft's development until December 2000.

Primary development was carried out by the Working Group's Task Group 1. The IEEE 802.16 Task Group 1 had the following officers:

Roger B. Marks, *Chair*
Carl Eklund, *MAC Chair*
Jay Klein, *PHY Chair*

The document finalization was overseen by an Editorial Committee. The IEEE 802.16 Task Group 1 Editorial Committee had the following officers:

Roger B. Marks, *Technical Editor*
Carl Eklund, Ken Stanwood, Stanley Wang, *Lead MAC Editors*
Jay Klein, Lars Lindh, *Lead PHY Editors*

Phil Guillemette, Wayne Hunter, Sergio Licardie, Ron Meyer, Vicente Quilez, Karl Stambaugh, Vladimir Yanover, and Juan Carlos Zúñiga made significant contributions to the draft as members of the Editorial Committee. Glen Sater served as editor in readying the document for Working Group Lettter Ballot, with contributions by Jeffrey Foerster. Earlier, Brian Petry and James Mollenauer served as editors. For a period, Jung Yee served as MAC Chair and John Liebetreu as PHY Vice Chair. Brian Petry served as editor of the Functional Requirements Document on which this standard was developed; George Fishel chaired the development of that document. Juan Carlos Zúñiga, Phil Guillemette, Ron Meyer, and Nico van Waes served as secretaries.

Technical contributions to the document came from many individuals. The Lead Editors were the primary architects of the technical development.

The following members of the IEEE 802.16 Working Group on Broadband Wireless Access participated in the Working Group Letter Ballot in which the draft of this standard was approved:

Song An
Jori Arrakoski
Arun Arunachalam
Eli Avivi
C. R. Baugh
Carlos Belfiore
Anader Benyamin-Seeyar
Carl Bushue
Baruch Buskila
Dean Chang
Naftali Chayat
Rémi Chayer
Mary Condie
José Costa
Bruce Currivan
Amos Dotan
Keith Doucet
Roger Durand
Brian Eidson
Carl Eklund
David Falconer
George Fishel
Adrian Florea
Jeff Foerster
Robert Foster
Avi Freedman
G. Jack Garrison
Conrad Grell
Phil Guillemette
Zion Hadad
Baruch Halachmi
Michael Hamilton
Baya Hatim
Srinath Hosur
Coleman Hum
Wayne Hunter
Eric Jacobsen
Hamadi Jamali
Jacob Jorgensen
Mika Kasslin
Brian Kiernan
John Kim
Itzik Kitroser
Allan Klein
Jay Klein
Demosthenes Kostas
Yigal Leiba
Barry Lewis
Sergio Licardie
John Liebetreu
Lars Lindh
Willie Lu
Fred Lucas
J. Scott Marin
Roger Marks
Andy McGregor
Ronald Meyer
Andrew Middleton
Apurva Mody
Jim Mollenauer
William Myers
Lou Olsen
Yunsang Park
Brian Petry
Wayne Pleasant
Moshe Ran
Stanley Reible
Valentine Rhodes
David Ribner
Gene Robinson
Walt Roehr
Durga Satapathy
Glen Sater
Vito Scaringi
Randall Schwartz
Menashe Shahar
Chet Shirali
George Stamatelos
Karl Stambaugh
Ken Stanwood
Michael Stewart
Paul Thompson
Subir Varma
Nico van Waes
Chao-Chun Wang
Bob Ward
Philip Whitehead
David G. Williams
Vladimir Yanover
Huanchun Ye
Chaoming Zeng
Juan Carlos Zúñiga

The following nonmembers also participated in the Working Group Letter Ballot:

Luis Contreras
Francisco Escrihuela
Moritz Harteneck
Babis Kalatzis
Antonis Karvelas
Tom Kolze
Gregorio Núñez
Subbu Ponnuswamy
Manuel Poza
Vicente Quilez
Andy Schiltz
Lei Wang
Stanley Wang

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

Larry Arnett
Tamer Beser
Tracy Black
Wesley Brodsky
James Carlo
Jose Costa
Jagadish Kumar Dasari
Thomas Dineen
Vern Dubendorf
Peter Ecclesine
Keng Fong
Simon Harrison
Vic Hayes
Hamadi Jamali
Stuart Kerry
Jay Klein
Timothy Lee
Arthur Light
Jenshan Lin
David Linton
Ahmad MahinFallah
James Scott Marin
Roger Marks
Peter Martini
George May
Roderick McMullin
Steve Methley
Vlad Mitlin
Roger Pandanda
Ken Peirce
Stephan Reis
Eugene Robinson
Walt Roehr
Jaideep Roy
Hiroyasu Shimizu
Chet Shirali
Fredrik Tufvesson
Joan Viaplana
Stanley S. Wang
Don Wright
Oren Yuen
Jim Zerbe

When the IEEE-SA Standards Board approved this standard on 6 December 2001, it had the following membership:

Donald N. Heirman, *Chair*
James T. Carlo, *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Mark D. Bowman
Gary R. Engmann
Harold E. Epstein
H. Landis Floyd
Jay Forster*
Howard M. Frazier
Ruben D. Garzon
James H. Gurney
Richard J. Holleman
Lowell G. Johnson
Robert J. Kennelly
Joseph L. Koepfinger*
Peter H. Lips
L. Bruce McClung
Daleep C. Mohla
James W. Moore
Robert F. Munzner
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Akio Tojo
Donald W. Zipse

*Member Emeritus

Also included is the following nonvoting IEEE-SA Standards Board liaison:

Alan Cookson, *NIST Representative*
Donald R. Volzka, *TAB Representative*

Jennifer McClain Longman
IEEE Standards Project Editor

Contents

List of Figures

List of Tables

IEEE Standard for
Local and metropolitan area networks

Part 16: Air Interface for Fixed Broadband Wireless Access Systems

1. Overview

1.1 Scope

This standard specifies the air interface, including the medium access control layer (MAC) and physical layer (PHY), of fixed point-to-multipoint broadband wireless access (BWA) systems providing multiple services. The MAC is structured to support multiple PHY specifications, each suited to a particular operational environment.

For the purposes of this document, a "system" consists of an IEEE Std 802.16™-2001 MAC and PHY implementation with at least one subscriber station communicating with a base station via a point-to-multipoint air interface, along with the interfaces to external networks and services transported by the MAC and PHY.

1.2 Purpose

This standard is intended to enable rapid worldwide deployment of innovative, cost-effective, and interoperable multivendor broadband wireless access products, to facilitate competition in broadband access by providing alternatives to wireline broadband access, to facilitate coexistence studies, to encourage consistent worldwide allocation, and to accelerate the commercialization of broadband wireless access spectrum.

The applications depend on the spectrum to be used. The primary bands of interest are as follows:

1.2.1 10–66 GHz licensed bands

The 10–66 GHz bands provide a physical environment where, due to the short wavelength, line of sight is required and multipath is negligible. The channels used in this physical environment are typically large. For example, channels 25 or 28 MHz wide are typical. With raw data rates in excess of 120 Mbit/s, this environment is well suited for point-to-multipoint access serving applications from small office/home office (SOHO) through medium to large office applications.

1.2.2 2–11 GHz

This work is in development under IEEE Standards Association Project P802.16a™.

1.3 Reference model

Figure 1 illustrates the reference model and scope of this standard.

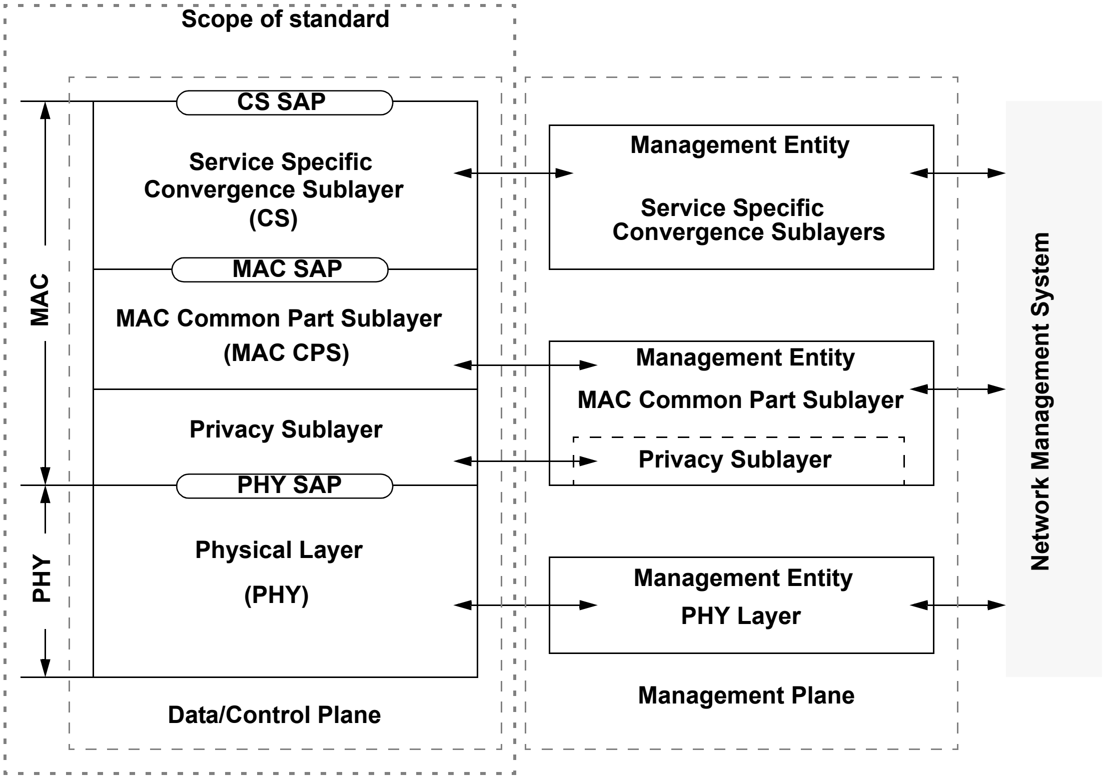


Figure 1—IEEE Std 802.16-2001 protocol layering, showing service access points (SAPs)

The MAC comprises three sublayers. The Service Specific Convergence Sublayer (CS) provides any transformation or mapping of external network data, received through the CS service access point (SAP), into MAC SDUs received by the MAC Common Part Sublayer (MAC CPS) through the MAC SAP. This includes classifying external network Service Data Units (SDUs) and associating them to the proper MAC service flow and Connection Identifier (CID). It may also include such functions as payload header suppression. Multiple CS specifications are provided for interfacing with various protocols. The internal format of the CS payload is unique to the CS, and the MAC CPS is not required to understand the format of or parse any information from the CS payload.

The MAC CPS provides the core MAC functionality of system access, bandwidth allocation, connection establishment, and connection maintenance. It receives data from the various CSs, through the MAC SAP, classified to particular MAC connections. Quality of Service (QoS) is applied to the transmission and scheduling of data over the PHY.

The MAC also contains a separate Privacy Sublayer providing authentication, secure key exchange, and encryption.

Data, PHY control, and statistics are transferred between the MAC CPS and the PHY via the PHY SAP.

The PHY may include multiple specifications, each appropriate to a particular frequency range and application. The various physical layer specifications supported are discussed in Clause 8.

2. References

This standard shall be used in conjunction with the following publications. When the following specifications are superseded by an approved revision, the revision shall apply.

ATM Forum Specification af-uni-0010.002, ATM User-Network Interface Specification, Version 3.1, September 1994.[1]

ATM Forum Specification af-sig-0061.000, ATM User-Network Interface (UNI) Signalling Specification, Version 4.0, July 1996.

ETSI EN 301 213-3, Fixed Radio Systems; Point-to-multipoint equipment; Point-to-multipoint digital radio systems in frequency bands in the range 24,25 GHz to 29,5 GHz using different access methods; Part 3: Time Division Multiple Access (TDMA) methods, Version 1.3.1, September 2001.[2]

FIPS 46-3, Data Encryption Standard (DES), October, 1999.[3]

FIPS 74, Guidelines for Implementing and Using the NBS Data Encryption Standard, April 1981.

FIPS 81, DES Modes of Operation, December 1980.

FIPS 180-1, Secure Hash Standard (SHS), April 1995.

FIPS 186-2, Digital Signature Standard (DSS), January 2000.

IEEE Std 802®-1990, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.[4, 5]

IEEE Std 802.1D™-1998, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges.[6]

IEEE Std 802.1Q™-1998, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.

IEEE Std 802.2™-1998 (ISO/IEC 8802-2: 1998), Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical Link Control.

IEEE Std 802.3™-2000 (ISO 8802-3), 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.

[1] ATM Forum publications are available from the ATM Forum at http://www.atmforum.com/.

[2] ESTI publications are available from the European Telecommunications Standards Institute at http://www.etsi.org/.

[3] FIPS publications are available from the National Technical Information Service (NTIS), U. S. Dept. of Commerce, 5285 Port Royal Road, Springfield, VA 22161 (http://www.ntis.gov/).

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

[5] IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://standards.ieee.org/).

[6] IEEE standards referred to in Clause 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

IEEE Std 802.16.2™-2001, IEEE Recommended Practice for Local and metropolitan area networks—Coexistence of Fixed Broadband Wireless Access Systems.

IETF RFC 791, “Internet Protocol,” J. Postel, September 1981.[7]

IETF RFC 868, “Time Protocol,” J. Postel, K. Harrenstien, May 1983.

IETF RFC 1042, “A Standard for the Transmission of IP Datagrams over IEEE 802 Networks,” J. Postel, J. Reynolds, February 1988.

IETF RFC 1123, “Requirements for Internet Hosts—Application and Support,” R. Braden, October 1989.

IETF RFC 1157, “A Simple Network Management Protocol (SNMP),” M. Schoffstall, M. Fedor, J. Davin, and J. Case, May 1990.

IETF RFC 2104, “HMAC: Keyed-Hashing for Message Authentication,” H. Krawczyk, M. Bellare, R. Canetti, February 1997.

IETF RFC 2131, “Dynamic Host Configuration Protocol,” R. Droms, March 1997.

IETF RFC 2132, “DHCP Options and BOOTP Vendor Extensions,” S. Alexander, and R. Droms, March 1997.

IETF RFC 2349, “TFTP Timeout Interval and Transfer Size Options,” G. Malkin and A. Harkin, May 1998.

IETF RFC 2459, “Internet X.509 Public Key Infrastructure Certificate and CRL Profile,” R. Housley, W. Ford, W. Polk, D. Solo, January 1999.

IETF RFC 2460, “Internet Protocol, Version 6 (IPv6) Specification,” S. Deering, R. Hinden, December 1998.

IETF RFC 2474, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” K. Nichols, S. Blake, F. Baker, D. Black, December 1998.

Internet Assigned Numbers Authority (IANA), “Protocol Numbers,” <http://www.iana.org/assignments/protocol-numbers>, June 2001.

ISO/IEC 8825, Information technology—Open Systems Interconnection—Specification of the Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), May 1999.[8]

ITU-T Recommendation X.690, Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER), December 1997.[9]

PKCS #1 v2.0, RSA Cryptography Standard, RSA Laboratories, October 1998 <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1>.

[7] IETF publications are available from the Internet Engineering Task Force at http://www.ietf.org/.

[8] ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Geneve 20, Switzerland/Suisse or the IEC Sales Department, Case Postale 131, 3, rue de Varembe, CH-1211, Geneve 20, Switzerland/Suisse. They are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th floor, New York, NY 10036, USA.

[9] ITU-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 standard, the following terms and definitions apply. The *IEEE 100, The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition [B4][10], should be referenced for terms not defined in this clause.

3.1 bandwidth stealing: The use, by a subscriber station operating on a *grant per subscriber station* basis, of a portion of the bandwidth allocated in response to a bandwidth request for a connection to send another bandwidth request rather than sending data. *See also*: **grant per subscriber station**.

3.2 base station (BS): A generalized equipment set providing connectivity, management, and control of the subscriber station.

3.3 Basic Connection: Connection that is established during initial subscriber station registration and that which is used to transport delay-intolerant MAC management messages.

3.4 broadband: Having instantaneous bandwidths greater than around 1 MHz and supporting data rates greater than about 1.5 Mbit/s.

3.5 broadband wireless access (BWA): Wireless access in which the connection(s) capabilities are broadband.

3.6 burst profile: Set of parameters that describe the uplink or downlink transmission properties associated with an interval usage code. Each profile contains parameters such as modulation type, forward error correction type, preamble length, guard times, etc. *See also*: **interval usage code**.

3.7 byte: Throughout this standard, one byte is 8 bits.

3.8 Channel Identifier (ChID): An identifier used to distinguish between multiple uplink channels, all of which are associated with the same downlink channel.

3.9 concatenation: The act of combining multiple medium access control (MAC) protocol data units (PDUs) into a single time division multiplex (TDM) or time division multiple access (TDMA) burst.

3.10 connection: A unidirectional mapping between base station and subscriber station medium access control (MAC) peers for the purpose of transporting a service flow's traffic. Connections are identified by a connection identifier (CID). All traffic is carried on a connection, even for service flows that implement connectionless protocols, such as internet protocol (IP). *See also*: **connection identifier**.

3.11 Connection Identifier (CID): A unidirectional, medium access control layer address that identifies a connection to equivalent peers in the medium access control layer of the base station and subscriber station. It maps to a *service flow identifier* (SFID), which defines the quality of service (QoS) parameters of the service flow associated with that connection. Security associations also exist between keying material and CIDs. *See also*: **service flow identifier**.

3.12 downlink: The direction from the base station to the subscriber station.

3.13 Downlink Channel Descriptor (DCD): A medium access control layer message that describes the physical layer characteristics of a downlink channel.

3.14 Downlink Interval Usage Code (DIUC): An interval usage code (IUC) specific to a downlink. *See also*: **interval usage code**.

[10] The numbers in brackets correspond to those of the bibliography in Annex A.

3.15 downlink map (DL-MAP): A medium access control layer message that defines burst start times for both time division multiplex and time division multiple access by a subscriber station on the downlink.

3.16 dynamic service: The set of messages and protocols that allow the base station and subscriber station to add, modify, or delete the characteristics of a service flow.

3.17 fixed wireless access: Wireless access application in which the location of the base station and subscriber station are fixed in location during operation.

3.18 frame: A structured data sequence of fixed duration used by some physical layer specifications. A frame may contain both an uplink subframe and a downlink subframe.

3.19 frequency division duplex (FDD): A duplex scheme in which uplink and downlink transmissions use different frequencies but are typically simultaneous.

3.20 grant per connection (GPC): A bandwidth allocation method in which grants are allocated to a specific connection within a subscriber station. Note that bandwidth requests are always made for a connection.

3.21 grant per subscriber station (GPSS): A bandwidth allocation method in which grants are aggregated for all connections within a subscriber station and are allocated to the subscriber station as that aggregate. Note that bandwidth requests are always made for a connection.

3.22 information element (IE): A component of the downlink or uplink map that defines the starting address associated with an interval usage code (IUC). *See also*: **interval usage code**.

3.23 Initial Ranging Connection Identifier: A well-defined connection identifier that is used by a subscriber station during the initial ranging process. This Connection Identifier (CID) is defined as constant value within the protocol since a subscriber station has no addressing information available until the initial ranging process is complete.

3.24 Interval Usage Code (IUC): A code identifying a particular burst profile that can be used by a downlink or uplink transmission interval.

3.25 management connection: A connection that is established during initial subscriber station registration that is used to transport delay-tolerant medium access control management messages.

3.26 minislot: A unit of uplink bandwidth allocation equivalent to n physical slots, where $n = 2^m$ and m is an integer ranging from 0 through 7.

3.27 multicast polling group: A group of zero or more subscriber stations that are assigned a multicast address for the purposes of polling.

3.28 packing: The act of combining multiple service data units from a higher layer into a single medium access control protocol data unit.

3.29 payload header suppression (PHS): The process of suppressing the repetitive portion of payload headers at the sender and restoring the headers at the receiver.

3.30 Payload Header Suppression Field (PHSF): A string of bytes representing the header portion of a protocol data unit in which one or more bytes are to be suppressed (i.e., a snapshot of the uncompressed protocol data unit header inclusive of suppressed and unsuppressed bytes).

3.31 Payload Header Suppression Index (PHSI): An 8-bit mask that indicates which bytes in the payload header suppression field to suppress and which bytes to not suppress.

3.32 Payload Header Suppression Size (PHSS): The length of the suppressed field in bytes. This value is equivalent to the number of bytes in the payload header suppression field and also the number of valid bits in the payload header suppression mask.

3.33 Payload Header Suppression Valid (PHSV): A flag that tells the sending entity to verify all bytes that are to be suppressed.

3.34 physical slot (PS): A unit of time, dependent on the physical layer specification, for allocating bandwidth

3.35 privacy key management protocol (PKM): A client/server model between the base station and subscriber station that is used to secure distribution of keying material.

3.36 protocol data unit (PDU): The data unit exchanged between peer entities of the same protocol layer. On the downward direction, it is the data unit generated for the next lower layer. On the upward direction, it is the data unit received from the previous lower layer (see Figure 2).

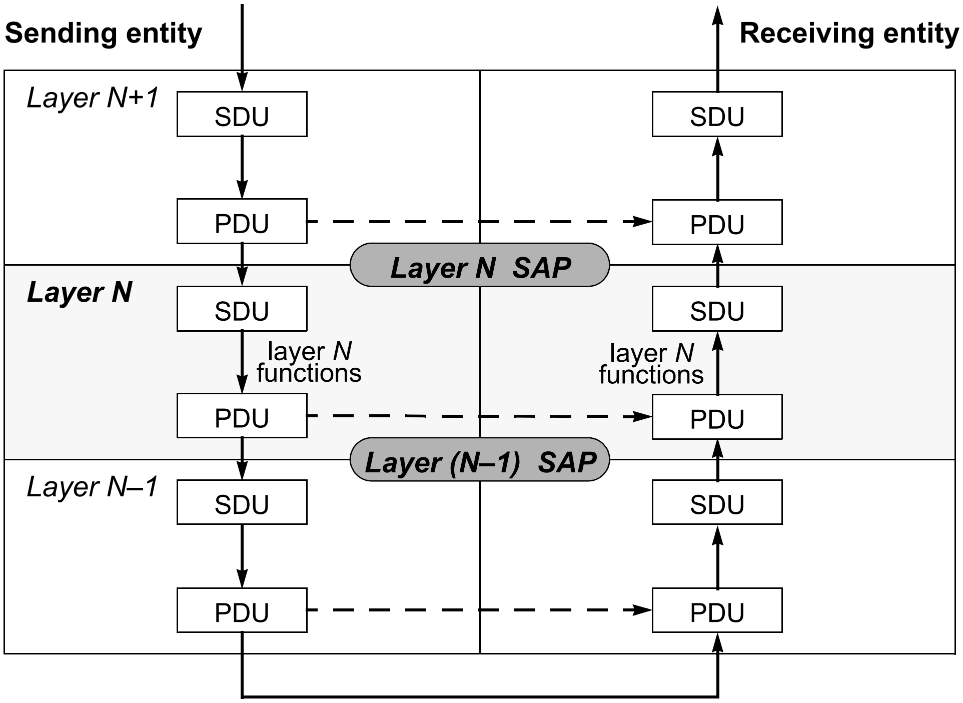


Figure 2—PDU and SDU in a protocol stack

3.37 security association (SA): The set of security information a base station and one or more of its client subscriber stations share in order to support secure communications. This shared information includes traffic encryption keys and cipher block chaining initialization vectors.

3.38 Security Association Identifier (SAID): An identifier shared between the base station and subscriber station that uniquely identifies a security association.

3.39 service access point (SAP): The point in a protocol stack where the services of a lower layer are available to its next higher layer.

3.40 service data unit (SDU): The data unit exchanged between two adjacent protocol layers. On the downward direction, it is the data unit received from the previous higher layer. On the upward direction, it is the data unit sent to the next higher layer (see Figure 2).

3.41 service flow (SF): A unidirectional flow of medium access control service data units on a connection that is provided a particular quality of service.

3.42 service flow class: A grouping of service flow properties to allow higher layer entities and external applications to request service flows with desired quality of service (QoS) parameters in a globally consistent way.

3.43 Service Flow Identifier (SFID): A 32-bit quantity that uniquely identifies a service flow to both the subscriber station and base station.

3.44 service flow name: An ASCII string that is used to reference a set of quality of service (QoS) parameters that (partially) define a service flow.

3.45 subscriber station (SS): A generalized equipment set providing connectivity between subscriber equipment and a base station.

3.46 time division duplex (TDD): A duplex scheme where uplink and downlink transmissions occur at different times but may share the same frequency.

3.47 time division multiple access (TDMA) burst: A contiguous portion of the uplink or downlink using physical layer parameters, determined by the downlink or uplink interval usage code, that remain constant for the duration of the burst. TDMA bursts are separated by preambles and are separated by gaps in transmission if subsequent bursts are from different transmitters.

3.48 time division multiplex (TDM) burst: A contiguous portion of a time division multiplex (TDM) data stream using physical layer parameters, determined by the downlink interval usage code, that remain constant for the duration of the burst. TDM bursts are not separated by gaps or preambles.

3.49 Transport Connection Identifier: A unique identifier taken from the connection identifier address space that uniquely identifies the transport connection.

3.50 transport connection: A connection used to transport user data.

3.51 type/length/value (TLV): A formatting scheme that adds a tag to each transmitted parameter containing the parameter type (and implicitly its encoding rules) and the length of the encoded parameter.

3.52 uplink: The direction from a subscriber station to the base station.

3.53 Uplink Channel Descriptor (UCD): A medium access control message that describes the physical layer characteristics of an uplink.

3.54 Uplink Interval Usage Code (UIUC): An interval usage code specific to an uplink.

3.55 uplink map (UL-MAP): A set of information that defines the entire access for a scheduling interval.

3.56 wireless access: End-user radio connection(s) to core networks.

4. Abbreviations and acronyms

3-DES	triple data encryption standard
AK	authorization key
ARP	address resolution protocol
ARQ	automatic repeat request
ATDD	adaptive time division duplexing
ATM	asynchronous transfer mode

BCC	block convolutional code
BE	best effort
BNI	base station network interface
BR	bandwidth request
BS	base station
BTC	block turbo code
BWA	broadband wireless access
C/(I+N)	carrier-to/(interference plus noise) ratio
C/I	carrier-to-interference ratio
C/N	carrier-to-noise ratio
CA	certification authority
CBC	cipher block chaining
CC	confirmation code
CCS	common channel signaling
CCV	clock comparison value
CEPT	European Conference of Postal and Telecommunications Administrations
CG	continuous grant
ChID	Channel Identifier
CID	Connection Identifier
CLP	Cell Loss Priority
CPS	common part sublayer
CRC	cyclic redundancy check
CS	convergence sublayer
DAMA	demand assigned multiple access
DCD	Downlink Channel Descriptor
DES	data encryption standard
DHCP	dynamic host configuration protocol
DIUC	Downlink Interval Usage Code
DL	downlink
DSA	dynamic service addition
DSC	dynamic service change
DSCP	differentiated services codepoint
DSD	dynamic service deletion
DSx	dynamic service addition, change, or deletion
EC	encryption control
ECB	electronic code book
EDE	encrypt-decrypt-encrypt
EIRP	effective isotropic radiated power
EKS	encryption key sequence
ETSI	European Telecommunications Standards Institute
EUI	extended unique identifier
EVM	error vector magnitude
FC	fragmentation control
FDD	frequency division duplex
FEC	forward error correction
FIPS	Federal Information Processing Standard
FSH	fragmentation subheader
FSN	fragment sequence number
GF	Galois field
GM	grant management
GPC	grant per connection
GPSS	grant per subscriber station
HCS	header check sequence
HEC	header error check

HMAC	Hashed Message Authentication Code
HT	header type
IANA	Internet Assigned Numbers Authority
IE	information element
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
ITU	International Telecommunications Union
IUC	Interval Usage Code
IWF	interworking function
KEK	key encryption key
LAN	local area network
LFSR	linear feedback shift registers
LLC	logical link control
LMDS	local multipoint distribution service
lsb	least significant bit
LSB	least significant byte
MAC	medium access control layer
MAN	metropolitan area network
MIB	management information base
MIC	message integrity check
MMDS	multichannel multipoint distribution service
MPEG	Moving Pictures Experts Group
MPLS	multiprotocol label switching
msb	most significant bit
MSB	most significant byte
NNI	network-to-network interface (or network node interface)
nrtPS	non-real-time polling service
OID	object identifier
PBR	piggyback request
PDH	plesiochronous digital hierarchy
PDU	protocol data unit
PHS	Payload Header Suppression
PHSF	Payload Header Suppression Field
PHSI	Payload Header Suppression Index
PHSM	Payload Header Suppression Mask
PHSS	Payload Header Suppression Size
PHSV	Payload Header Suppression Valid
PHY	physical layer
PKM	privacy key management
PM	poll-me bit
PMD	physical medium dependent
ppm	parts per million
PPP	Point-to-Point Protocol
PS	physical slot
PSH	Packing Subheader
PTI	Payload Type Indicator
PVC	Permanent Virtual Circuit
QAM	quadrature amplitude modulation
QoS	Quality of Service
QPSK	quadrature phase-shift keying
RS	Reed–Solomon
RSSI	receive signal strength indicator
rtPS	real-time polling service

Rx	reception
SA	security association
SAID	security association identifier
SAP	service access point
SCTE	Society of Cable Telecommunications Engineers
SDH	Synchronous Digital Hierarchy
SDU	service data unit
SF	service flow
SFID	Service Flow Identifier
SHA	secure hash algorithm
SI	slip indicator
SNMP	Simple Network Management Protocol
SS	subscriber station
SVC	Switched Virtual Circuit
TC	transmission convergence sublayer
TCP	Transmission Control Protocol
TDD	time division duplex
TDM	time division multiplex
TDMA	time division multiple access
TEK	traffic encryption key
TFTP	Trivial File Transfer Protocol
TLV	type-length-value
Tx	transmission
UCD	Uplink Channel Descriptor
UDP	User Datagram Protocol
UGS	unsolicited grant service
UIUC	Uplink Interval Usage Code
UL	uplink
UNI	user-to-network interface
UTC	Coordinated Universal Time
VC	virtual channel
VCI	virtual channel identifier
VLAN	virtual local area network
VP	virtual path
VPI	Virtual Path Identifier
XOR	exclusive or

5. Service specific convergence sublayer

The service specific convergence sublayer (CS) resides on top of the MAC CPS and utilizes, via the MAC SAP, the services provided by the MAC CPS (see Figure 1). The CS performs the following functions:

— accepting higher-layer PDUs from the higher layer

— performing classification of higher-layer PDUs

— processing (if required) the higher-layer PDUs based on the classification

— delivering CS PDUs to the appropriate MAC SAP

— receiving CS PDUs from the peer entity

Currently, two CS specifications are provided: the asyncronous transfer mode (ATM) CS and the packet CS. Other CSs may be specified in the future.

5.1 ATM convergence sublayer

The ATM CS is a logical interface that associates different ATM services with the MAC CPS SAP. The ATM CS accepts ATM cells from the ATM layer, performs *classification* and, if provisioned, *payload header suppression* (PHS), and delivers CS PDUs to the appropriate MAC SAP.

5.1.1 Convergence sublayer service definition

The ATM CS is specifically defined to support the convergence of PDUs generated by the ATM layer protocol of an ATM network. Since ATM cell streams are generated according to the ATM standards, no ATM CS service primitive is required.

5.1.2 Data/Control plane

5.1.2.1 PDU formats

The ATM CS PDU shall consist of an ATM CS PDU Header, defined in Table 1, and the ATM CS PDU payload. The ATM CS PDU payload shall be equal to the ATM cell payload. The ATM CS PDU is illustrated in Figure 3.

Table 1—ATM CS PDU Header

Syntax	Size	Notes
ATM_CS_PDU_Header (){		
if (no PHS) {		
ATM_Header	40 bits	The full ATM cell header
}		
else if (VP switched) {		
PTI	3 bits	From the ATM cell header
CLP	1 bit	From the ATM cell header
reserved	4 bits	
VCI	16 bits	From the ATM cell header
}		
else (VC switched) {		
PTI	3 bits	From the ATM cell header
CLP	1 bit	From the ATM cell header
reserved	4 bits	
}		
}		

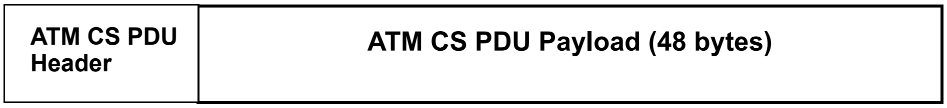


Figure 3—ATM CS PDU format

5.1.2.2 Classification

An ATM connection, which is uniquely identified by a pair of values of Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI), is either Virtual Path (VP) switched or Virtual Channel (VC) switched. In VP-switched mode, all VCIs within one single incoming VPI are automatically mapped to that of an outgoing VPI. In VC-switched mode, input VPI/VCI values are individually mapped to output VPI/VCI values. Thus, when performing PHS, the ATM CS differentiates these two types of connections and performs the suppression accordingly.

A classifier is a set of matching criteria applied to each ATM cell entering the ATM CS. It consists of some ATM cell matching criteria, such as VPI and VCI, and a reference to a CID. If an ATM cell matches the specified matching criteria, it is delivered to the MAC SAP for delivery on the connection identified by the CID.

5.1.2.2.1 VP-switched mode

For VP-switched mode, the VPI field, 12 bits for a network-to-network interface (NNI) and 8 bits for a user-to-network interface (UNI), is mapped to the 16-bit CID for the MAC connection on which it is transported. Since the QoS and category of service parameters for the connection are set at connection establishment, this mapping of VPI to CID guarantees the correct handling of the traffic by the MAC sublayer.

5.1.2.2.2 VC-switched mode

For VC switched mode, the VPI and VCI fields, 28 bits total for an NNI and 24 bits total for a UNI, are mapped to the 16-bit CID for the MAC connection on which it is transported. Since the QoS and category of service parameters for the connection are set at connection establishment, this mapping of VPI and VCI to CID guarantees the correct handling of the traffic by the MAC sublayer. Note that the full range of VPI/VCI combinations (up to 2^{28} for NNI and 2^{24} for UNI) cannot be simultaneously supported in this mode.

5.1.2.3 Payload header suppression

In payload header suppression (PHS), a repetitive portion of the payload headers of the CS SDUs is suppressed by the sending entity and restored by the receiving entity. On the downlink, the sending entity is the ATM CS on the base station (BS) and the receiving entity is the ATM CS on the SS. On the uplink, the sending entity is the ATM CS on the SS, and the receiving entity is the ATM CS on the BS. To further save bandwidth, multiple ATM cells (with or without PHS) that share the same CID may be packed and carried by a single MAC CPS PDU. Note that when PHS is turned off, no part of any ATM cell header including Header Error Check (HEC) field shall be suppressed. This provides an option for protecting the integrity of the cell header. Whether or not PHS is applied to an ATM connection is signaled in the DSA-REQ message at the connection’s creation. Similarly, the VPI (for VP-switched connections) or the VPI/VCI (for VC-switched connections) is also signaled in the classifier settings of the DSA-REQ message at connection creation.

5.1.2.3.1 PHS for VP-switched ATM connections

In VP switched mode, the VPI is mapped to a CID. This allows the disposal of the remainder of the ATM cell header except for the VCI, payload type indicator (PTI), and cell loss priority (CLP) fields. These fields shall be encapsulated in the CS PDU header.

Figure 4 shows a CS PDU containing a single VP-switched ATM cell with the cell header suppressed and the format of the ATM CS PDU Header for VP-switched ATM connections.

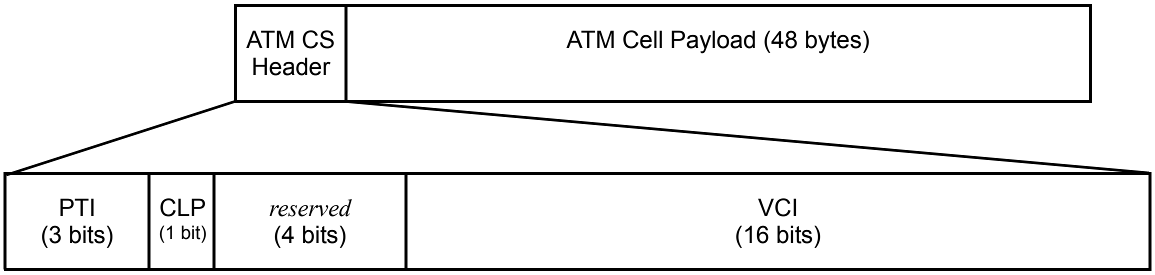


Figure 4—CS PDU format for VP-switched ATM connections

5.1.2.3.2 PHS for VC-switched ATM connections

In VC-switched mode, the VPI/VCI combination is mapped to a CID. This allows the disposal of the remainder of the ATM cell header except for the PTI and CLP fields. These fields shall be encapsulated in the CS PDU Header.

Figure 5 shows a CS PDU containing a single VC-switched ATM cell with the cell header suppressed and the format of the ATM CS PDU header for VC-switched ATM connections.

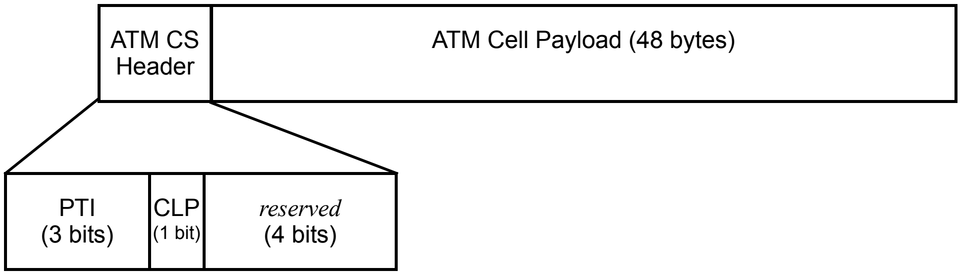


Figure 5—CS PDU format for VC-switched ATM connections

5.1.2.4 Signaling procedure

ATM interfaces support three types of connections, switched virtual circuit (SVC), permanent virtual circuit (PVC), and soft permanent virtual circuit (soft PVC). SVCs are established and terminated dynamically on demand by the use of signaling. The word “permanent” signifies that the circuit is established administratively. Although both PVC and soft PVC are established administratively, PVCs are established by provisioning process, and soft PVCs are established by the use of signaling.

ATM networks use common channel signaling (CCS), where signaling messages are carried over a connection completely independent of user connections and where one signaling channel can carry signaling messages for a number of user connections. Per nonassociated signaling (ATM as-sig-0061.000), by default, the signaling channel on VPI=0 controls all VPs on the same physical interface. In other words, except when the optional proxy signaling capability (Annex 2 of ATM as-sig-0061.000) or when the optional Virtual UNI capability (Annex 8 of ATM as-sig-0061.000) is used, the signaling channel is identified by VPI=0 and VCI=5. Note that this specification does not support associated signaling (ATM af-uni-0010.002), where VCI=5 of each VP is used as the signaling channel for all VCs on the same VP. In addition, this specification does not support either proxy signaling or virtual UNI.

To establish an SVC, it is the responsibility of the calling party to initiate the signaling procedure by issuing the appropriate signaling messages. Either end can establish or release the SVC. Details on how to use these signaling messages are available in ATM as-sig-0061.000. It shall be the responsibility of the implementation of the BS to map ATM signaling messages to corresponding MAC CPS service primitives.

To establish a soft PVC, the network management system provisions one end of the soft PVC with the address identifying the egress ATM interface of the ATM network. The calling end has the responsibility for establishing and releasing the connection. It is also the responsibility of the calling party (if necessary) to reestablish the connection in case of switching system or link failure. It shall be the responsibility of the implementation of the BS to map ATM signaling messages to corresponding MAC CPS service primitives.

On the downlink direction, the signaling starts at an “end user” of the ATM backhaul network that implements an ATM UNI and terminates at the BS that shall implement either an ATM UNI or an ATM NNI. The signaling may be mapped by an interworking function (IWF) and extended to some user network on the SS-side. On the uplink direction, the signaling starts at the ATM interface of the BS and ends at the ATM UNI of an “end user.” In addition, the signaling may be originated by an “end user” of some user network and mapped by the IWF. Note that mapping of data units carried by the air link shall be limited to only cell-level convergence (5.1.2.2). If required by a user network, other levels of mappings (e.g., the convergence of, say, an Ethernet packet to ATM cells) shall be handled by the user network’s IWF exclusively.

During the provisioning process, each SS joining the IEEE Std 802.16-2001 system shall request a dedicated CID as the signaling connection corresponding to the CCS connection used by ATM networks. Any CID provisioned for this purpose shall not be dynamically changed or terminated. Each IEEE Std 802.16-2001 system shall provision a set of CIDs for this purpose.

5.2 Packet convergence sublayer

The packet CS resides on top of the IEEE Std 802.16-2001 MAC CPS layer. The CS performs the following functions, utilizing the services of the MAC sublayer:

a) Classification of the higher-layer protocol PDU into the appropriate connection
b) Suppression of payload header information (optional)
c) Delivery of the resulting CS PDU to the MAC SAP associated with the service flow for transport to the peer MAC SAP
d) Receipt of the CS PDU from the peer MAC SAP
e) Rebuilding of any suppressed payload header information (optional)

The sending CS is responsible for delivering the MAC SDU to the MAC SAP. The MAC is responsible for delivery of the MAC SDU to peer MAC SAP in accordance with the QoS, fragmentation, concatenation and other transport functions associated with a particular connection’s service flow characteristics. The receiving CS is responsible for accepting the MAC SDU from the peer MAC SAP and delivering it to a higher-layer entity.

The packet CS is used for transport for all packet-based protocols such as internet protocol (IP), point-to-point protocol (PPP), and IEEE Std 802.3 (Ethernet).

5.2.1 MAC SDU format

Higher-layer PDUs shall be encapsulated in the MAC SDU format as illustrated in Figure 6. For some payload protocols, each payload consists of an 8-bit payload header suppression index (PHSI) field followed by the actual payload. Other protocols map the higher layer PDU directly to the MAC SDU. A value of zero in the PHSI indicates no payload header suppression has been applied to the PDU. Otherwise, the value in the index identifies the rules for suppression. This index is mapped to equivalent rules at BS and SS peers to allow for reconstruction of suppressed information.

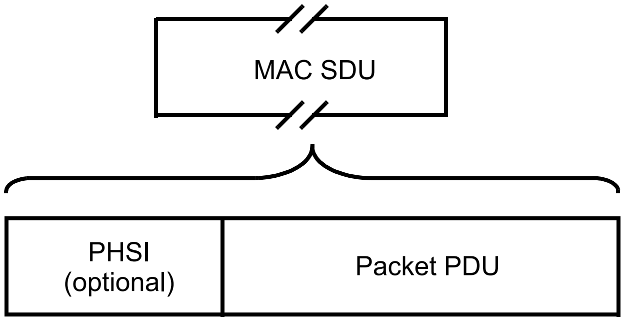


Figure 6—MAC SDU Format

5.2.2 Classification

Classification is the process by which a MAC SDU is mapped onto a particular connection for transmission between MAC peers. The mapping process associates a MAC SDU with a connection, which also creates an association with the service flow characteristics of that connection. This process facilitates the delivery of MAC SDUs with the appropriate QoS constraints.

A classifier is a set of matching criteria applied to each packet entering the IEEE Std 802.16-2001 network. It consists of some protocol-specific packet matching criteria (destination IP address, for example), a classifier priority, and a reference to a CID. If a packet matches the specified packet matching criteria, it is then delivered to the SAP for delivery on the connection defined by the CID. The service flow characteristics of the connection provide the QoS for that packet.

Several classifiers may each refer to the same service flow. The classifier priority is used for ordering the application of classifiers to packets. Explicit ordering is necessary because the patterns used by classifiers may overlap. The priority need not be unique, but care shall be taken within a classifier priority to prevent ambiguity in classification. Downlink classifiers are applied by the BS to packets it is transmitting and uplink classifiers are applied at the SS. Figure 7 and Figure 8 illustrate the mappings discussed above.

It is possible for a packet to fail to match the set of defined classifiers. In this case, the CS may either associate the packet with a default CID or discard the packet. The action taken is vendor specific.

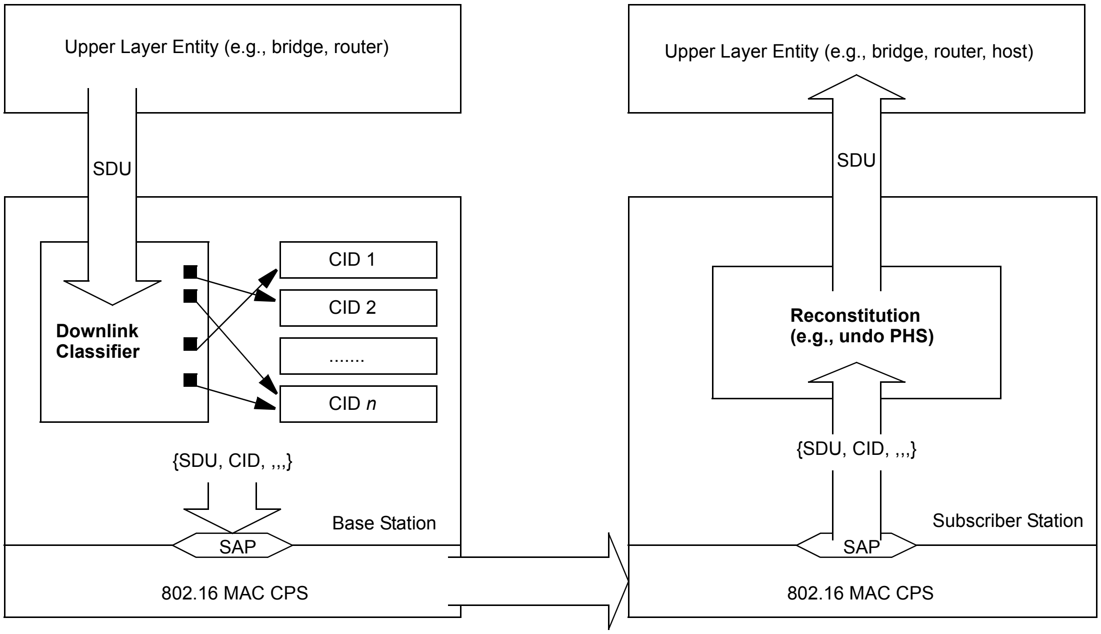


Figure 7—Classification and CID mapping (BS to SS)

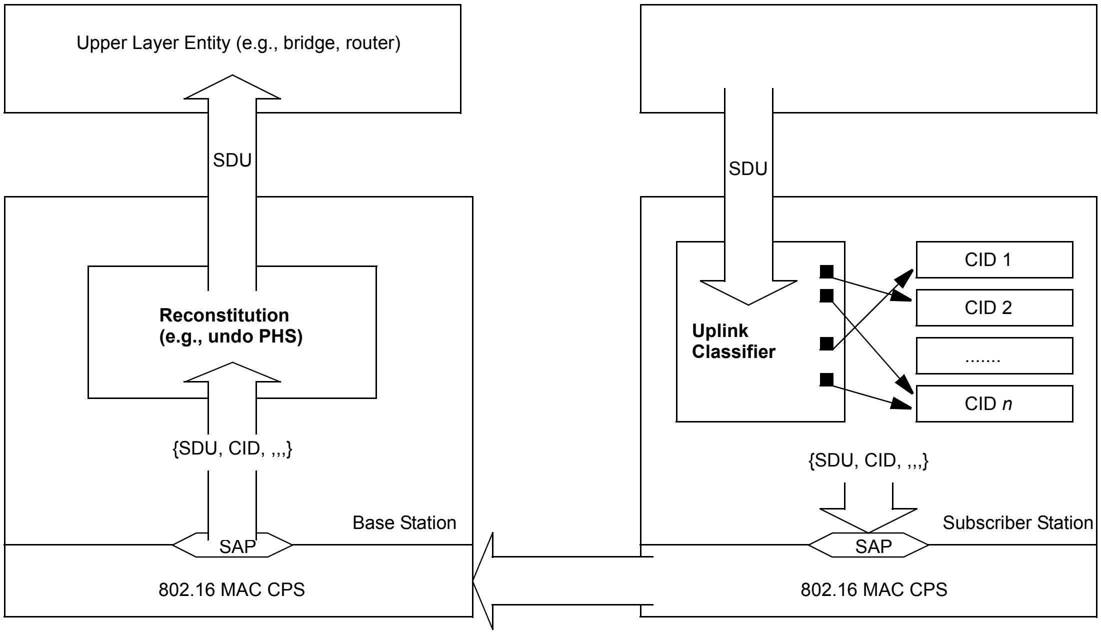


Figure 8—Classification and CID mapping (SS to BS)

5.2.3 Classification within the CS

SS and BS packet classification consists of multiple classifiers. Each classifier contains a priority field which determines the search order for the classifier. The highest priority classifier shall be applied first. If a classifier is found in which all parameters match the packet, the classifier shall forward the packet to the corresponding connection(s). If no classifier is found in which all parameters match the packet then the packet is delivered under vendor or operator specific conditions. Two actions may be performed: the packet may be delivered using a “default” connection, or the packet may be discarded.

Classifiers can be added to the table either via network management operations or via dynamic operations (dynamic signaling, IEEE Std 802.16-2001 MAC sublayer service interface). Simple Network Management Protocol (SNMP)-based operations can view classifiers that are added via dynamic operations, but they shall not modify or delete classifiers that are created by dynamic operations. The format for classification table parameters defined in the configuration file or dynamic signaling message is contained in 11.4.9.3.

Typically, an outgoing user data packet is submitted by an upper-layer protocol (such as the forwarding bridge of an SS) for transmission on the MAC interface. The packet is compared against a set of classifiers. The matching classifier for the packet identifies the corresponding service flow via the service flow ID (SFID). In the case where more than one classifier matches the packet, the highest priority classifier is chosen.

The classifier matching a packet may be associated with a payload header suppression rule. A PHS rule provides details on how header bytes of a packet PDU may be omitted, replaced with a PHSI for transmission, and subsequently regenerated at the receiving end. PHS rules are indexed by the combination of {SFID, PHSI}. When a service flow is deleted, all classifiers and any associated PHS rules referencing it shall also be deleted.

5.2.4 Payload header suppression

In payload header suppression (PHS), a repetitive portion of the payload headers of the higher layer is suppressed in the MAC SDU by the sending entity and restored by the receiving entity. On the uplink, the sending entity is the SS and the receiving entity is the BS. On the downlink, the sending entity is the BS and the receiving entity is the SS. Each MAC SDU is prefixed with a PHSI, which references the payload header suppression field (PHSF).

The sending entity uses classifiers to map packets into a service flow. The classifier uniquely maps packets to its associated PHS Rule. The receiving entity uses the CID and the PHSI to restore the PHSF. Once a PHSF has been assigned to a PHSI, it shall not be changed. To change the value of a PHSF on a service flow, a new PHS rule shall be defined, the old rule is removed from the service flow, and the new rule is added. When a classifier is deleted, any associated PHS rule shall also be deleted.

PHS has a payload header suppression valid (PHSV) option to verify or not verify the payload header before suppressing it. PHS has also a payload header suppression mask (PHSM) option to allow select bytes not to be suppressed. This is used for sending bytes that change, such as IP sequence numbers, while still suppressing bytes that do not change.

The BS shall assign all PHSI values just as it assigns all CID values. Either the sending or the receiving entity shall specify the PHSF and the payload header suppression size (PHSS). This provision allows for preconfigured headers or for higher-level signaling protocols outside the scope of this specification to establish cache entries. PHS is intended for unicast service and is not defined for multicast service.

It is the responsibility of the higher-layer service entity to generate a PHS Rule which uniquely identifies the suppressed header within the service flow. It is also the responsibility of the higher-layer service entity to guarantee that the byte strings being suppressed are constant from packet to packet for the duration of the active service flow.

5.2.4.1 PHS operation

SS and BS implementations are free to implement PHS in any manner as long as the protocol specified in this subclause is followed. Figure 9 illustrates the following procedure.

A packet is submitted to the packet CS. The SS applies its list of Classifier rules. A match of the rule shall result in an Uplink Service Flow, CID, and a PHS Rule. The PHS Rule provides PHSF, PHSI, PHSM, PHSS, and PHSV. If PHSV is set or not present, the SS shall compare the bytes in the packet header with the bytes in the PHSF that are to be suppressed as indicated by the PHSM. If they match, the SS shall suppress all the bytes in the Uplink PHSF except the bytes masked by PHSM. The SS shall then prefix the PDU with the PHSI and present the entire MAC SDU to the MAC SAP for transport on the uplink.

When the packet is received by the BS, the BS shall determine the associated CID by examination of the generic MAC header. The BS sends the PDU to the MAC SAP associated with that CID. The receiving packet CS uses the CID and the PHSI to look up PHSF, PHSM, and PHSS. The BS reassembles the packet and then proceeds with normal packet processing. The reassembled packet contains bytes from the PHSF. If verification was enabled, then the PHSF bytes equal the original header bytes. If verification was not enabled, then there is no guarantee that the PHSF bytes match the original header bytes.

A similar operation occurs on the downlink. The BS applies its list of Classifiers. A match of the Classifier shall result in a Downlink Service Flow and a PHS Rule. The PHS Rule provides PHSF, PHSI, PHSM, PHSS, and PHSV. If PHSV is set or not present, the BS shall verify the Downlink Suppression Field in the packet with the PHSF. If they match, the BS shall suppress all the bytes in the Downlink Suppression Field except the bytes masked by PHSM. The BS shall then prefix the PDU with the PHSI and present the entire MAC SDU to the MAC SAP for transport on the downlink.

The SS shall receive the packet based upon the CID Address filtering within the MAC. The SS receives the PDU and then sends it to the CS. The CS then uses the PHSI to lookup PHSF, PHSM, and PHSS. The SS reassembles the packet and then proceeds with normal packet processing.

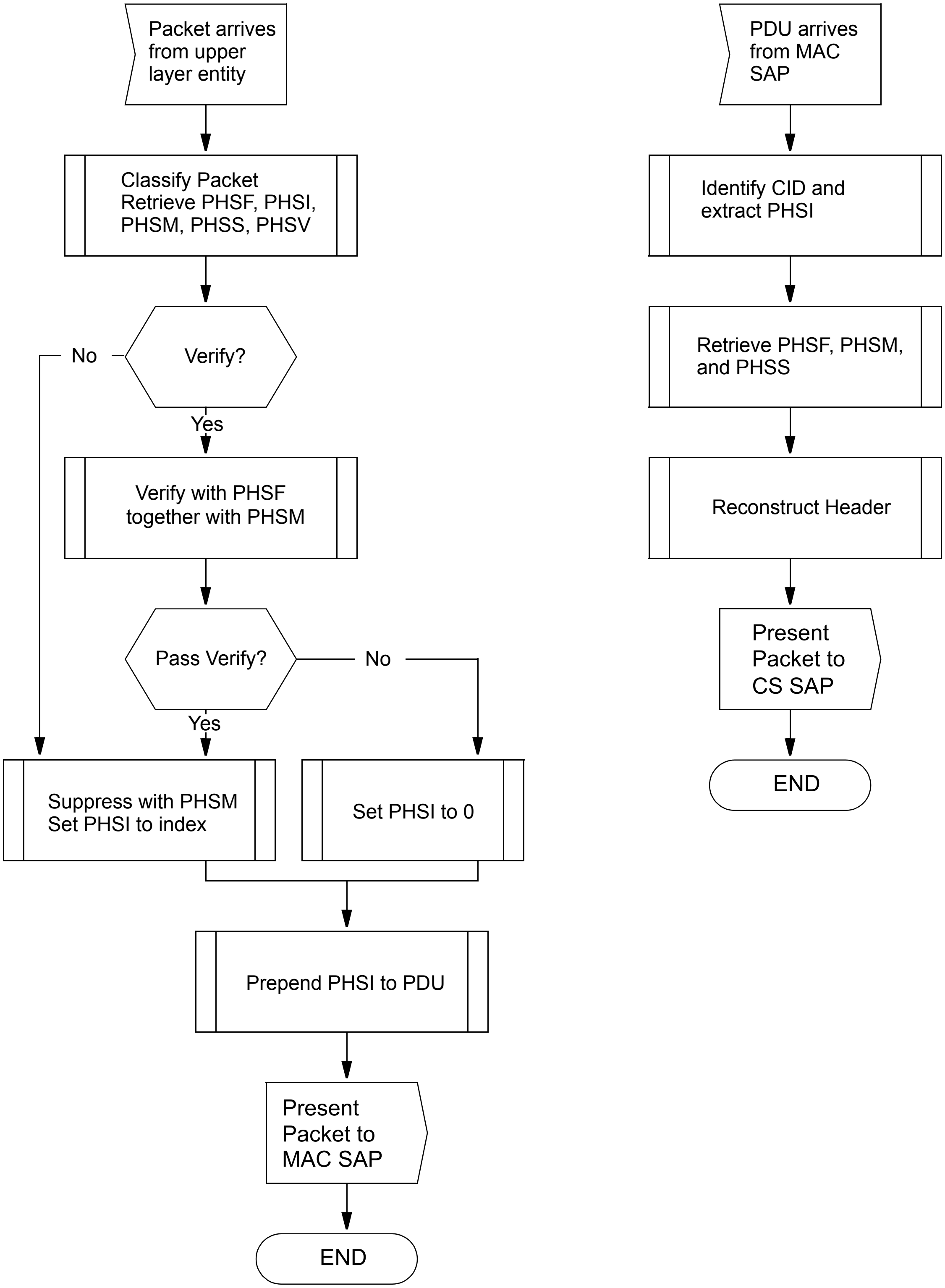

Packet arrives from upper layer entity
Classify Packet Retrieve PHSF, PHSI, PHSM, PHSS, PHSV
Verify?
No
Yes
Verify with PHSF together with PHSM
Pass Verify?
No
Yes
Suppress with PHSM Set PHSI to index
Set PHSI to 0
Prepend PHSI to PDU
Present Packet to MAC SAP
END
PDU arrives from MAC SAP
Identify CID and extract PHSI
Retrieve PHSF, PHSM, and PHSS
Reconstruct Header
Present Packet to CS SAP
END

Figure 9—Payload header suppression operation

Figure 10 demonstrates packet suppression and restoration when using PHS masking. Masking allows only bytes that do not change to be suppressed. Note that the PHSF and PHSS span the entire suppression field, included suppressed and unsuppressed bytes.

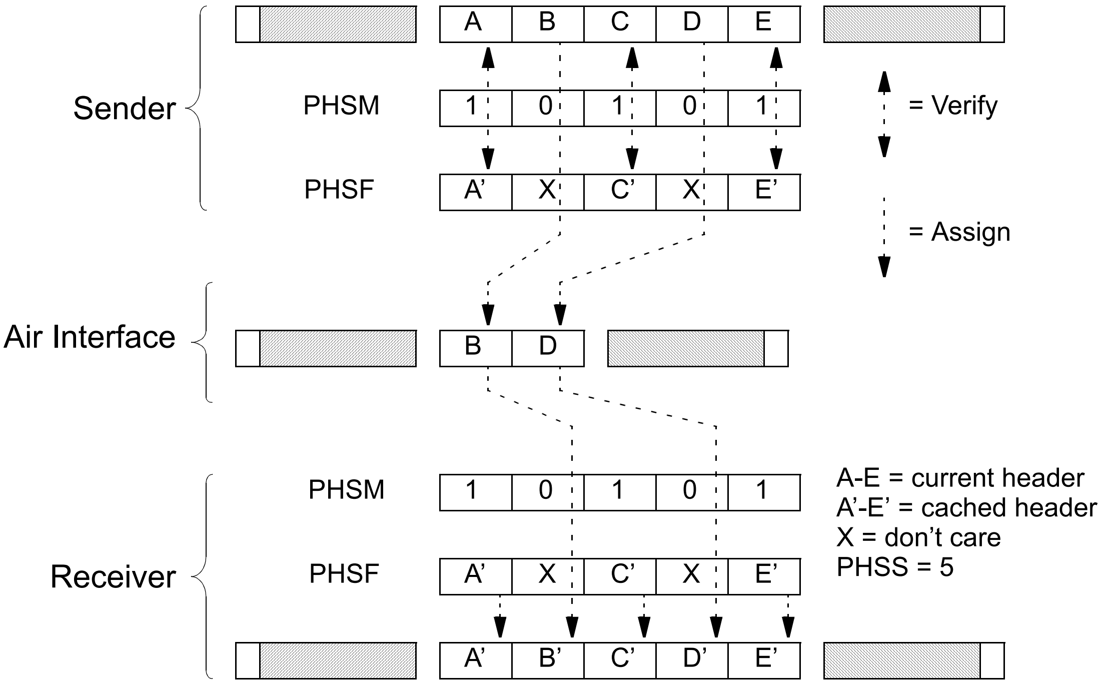


Figure 10—Payload header suppression with masking

5.2.4.2 PHS signaling

PHS requires the creation of the following three objects:

a) Service flow

b) Classifier

c) PHS rule

These three objects may be created either simultaneously or in separate message flows.

PHS Rules are created with dynamic service addition (DSA), or dynamic service change (DSC) messages. The BS shall define the PHSI when the PHS Rule is created. PHS rules are deleted with the DSC or dynamic service deletion (DSD) messages. The SS or BS may define the PHSS and PHSF. To change the value of a PHSF on a service flow, a new PHS rule shall be defined, the old rule is removed from the service flow, and the new rule is added.

Figure 11 shows the two ways to signal the creation of a PHS rule.

It is possible to partially specify a PHS rule (in particular the size of the rule) at the time a service flow is created. As an example, it is likely that, when a service flow is first provisioned, the header fields to be suppressed will be known. The values of some of the fields [e.g., IP addresses, user datagram protocol (UDP) port numbers, etc.] may be unknown and would be provided in a subsequent DSC as part of the activation of the service flow (using the "Set PHS Rule" DSC Action). If the PHS rule is being defined in more than one step, each step, whether it is a DSA or DSC message, shall contain both the service flow ID (or reference) and a PHS index to uniquely identify the PHS rule being defined.

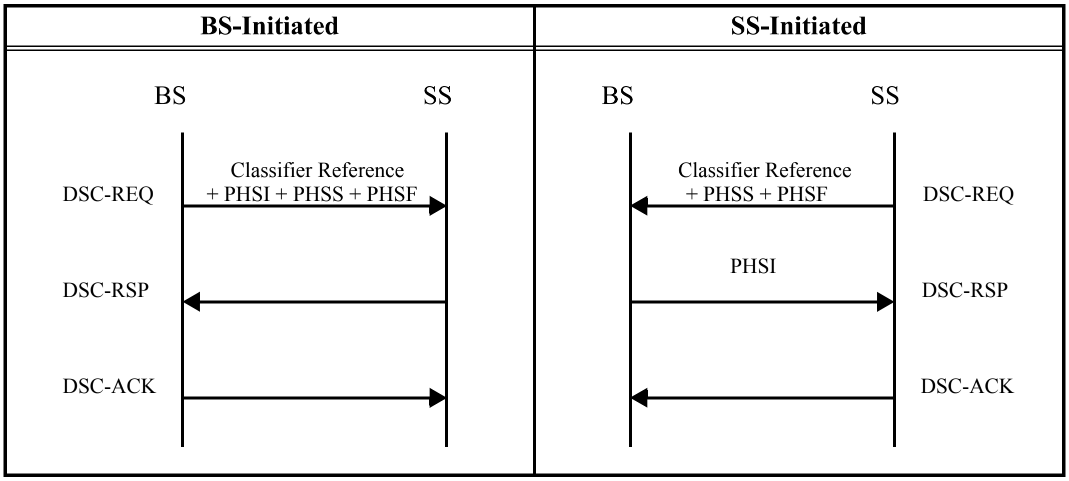


Figure 11—Payload header suppression signaling example

5.2.5 IEEE Std 802.3/Ethernet specific part

5.2.5.1 IEEE Std 802.3/Ethernet CS PDU format

The IEEE Std 802.3/Ethernet PDUs are mapped to MAC SDUs according to Figure 12 (when header suppression is not applied) or Figure 13 (with header suppression).

PHSI=0	IEEE 802.3/Ethernet PDU

Figure 12—IEEE 802.3/Ethernet CS PDU format without header suppression

PHSI≠0	Header-Suppressed IEEE 802.3/Ethernet PDU

Figure 13—IEEE 802.3/Ethernet CS PDU format with header suppression

5.2.5.2 IEEE Std 802.3/Ethernet CS classifiers

The following parameters are relevant for IEEE Std 802.3/Ethernet CS classifiers:

Logical link control (LLC) classification parameters—zero or more of the LLC classification parameters (destination MAC address, source MAC address, Ethertype/SAP).

For IP over IEEE 802.3/Ethernet, IP headers may be included in classification. In this case, the IP classification parameters (11.4.9.3.6.2—11.4.9.3.6.7) are allowed.

5.2.6 IEEE Std 802.1Q-1998 VLAN specific part

This CS shall be employed when IEEE Std 802.1Q-1998 tagged VLAN frames are to be carried over the IEEE Std 802.16-2001 network.

5.2.6.1 IEEE Std 802.1Q-1998 VLAN CS PDU format

The format of the IEEE Std 802.1Q-1998 VLAN CS PDU shall be as shown in Figure 14.

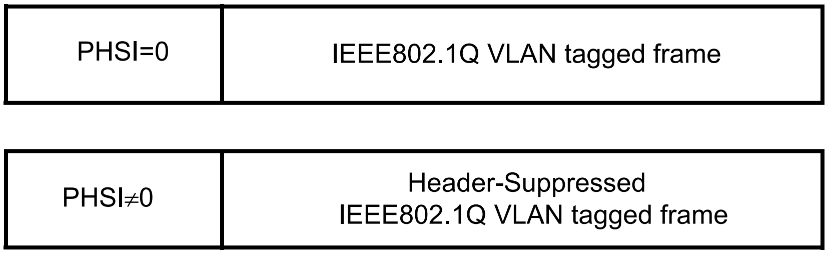


Figure 14—IEEE 802.1Q VLAN CS PDU format

5.2.6.2 IEEE Std 802.1Q-1998 CS classifiers

The following parameters are relevant for IEEE Std 802.1Q-1998 CS classifiers:

LC classification parameters—zero or more of the LLC classification parameters (Destination MAC address, source MAC address, Ethertype/SAP).

IEEE Std 802.1D-1998 Parameters—zero or more of the IEEE classification parameters (IEEE Std 802.1D-1998 Priority Range, IEEE Std 802.1Q-1998 VLAN ID).

For IP over IEEE Std 802.1Q-1998 VLAN, IP headers may be included in classification. In this case, the IP classification parameters (11.4.9.3.6.2—11.4.9.3.6.7) are allowed.

5.2.7 IP specific part

This subclause applies when IP (IETF RFC 791, IETF RFC 2460) is carried over the IEEE Std 802.16-2001 network.

5.2.7.1 IP CS PDU format

The format of the IP CS PDU shall be as shown in Figure 15 (when header suppression is not applied) or Figure 16 (with header suppression).

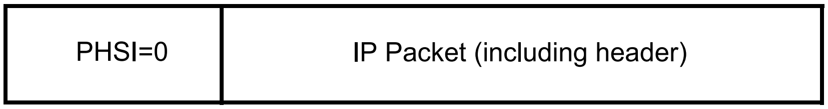


Figure 15—IP CS PDU format IEEE 802.3/Ethernet CS PDU format without header suppression

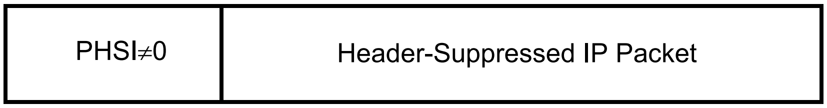


Figure 16—IP CS PDU format IEEE 802.3/Ethernet CS PDU format with header suppression

5.2.7.2 IP classifiers

IP classifiers operate on the fields of the IP header and the transport protocol. The parameters (11.4.9.3.6.2—11.4.9.3.6.7) may be used in IP classifiers.

6. MAC common part sublayer (CPS)

A network that utilizes a shared medium requires a mechanism to efficiently share it. A two-way point-to-multipoint wireless network is a good example of a shared medium; here the medium is the space through which the radio waves propagate.

The downlink, from the base station (BS) to the user, operates on a point-to-multipoint basis. The IEEE Std 802.16-2001 wireless link operates with a central BS and a sectorized antenna which is capable of handling multiple independent sectors simultaneously. Within a given frequency channel and antenna sector, all stations receive the same transmission, or parts thereof. The base station is the only transmitter operating in this direction, so it transmits without having to coordinate with other stations, except for the overall time division duplexing that may divide time into uplink and downlink transmission periods. It broadcasts to all stations in the sector (and frequency); stations check the address in the received messages and retain only those addressed to them.

In the other direction, the user stations share the uplink to the BS on a demand basis. Depending on the class of service utilized, the SS may be issued continuing rights to transmit, or the right to transmit may be granted by the BS after receipt of a request from the user.

In addition to individually addressed messages, messages may also be sent on multicast connections (control messages and video distribution are examples of multicast applications) as well as broadcast to all stations.

Within each sector, users adhere to a transmission protocol that controls contention between users and enables the service to be tailored to the delay and bandwidth requirements of each user application. This is accomplished through five different types of uplink scheduling mechanisms. These are implemented using unsolicited bandwidth grants, polling, and contention procedures. Mechanisms are defined in the protocol to allow vendors to optimize system performance using different combinations of these bandwidth allocation techniques while maintaining consistent inter-operability definitions. For example, contention may be used to avoid the individual polling of SSs which have been inactive for a long period of time.

The use of polling simplifies the access operation and guarantees that applications receive service on a deterministic basis if it is required. In general, data applications are delay tolerant, but real-time applications like voice and video require service on a more uniform basis and sometimes on a very tightly-controlled schedule.

The MAC is connection-oriented. For the purposes of mapping to services on SSs and associating varying levels of QoS, all data communications are in the context of a connection. Service flows may be provisioned when an SS is installed in the system. Shortly after SS registration, connections are associated with these service flows (one connection per service flow) to provide a reference against which to request bandwidth. Additionally, new connections may be established when a customer's service needs change. A connection defines both the mapping between peer convergence processes that utilize the MAC and a service flow. The service flow defines the QoS parameters for the PDUs that are exchanged on the connection.

The concept of a service flow on a connection is central to the operation of the MAC protocol. Service flows provide a mechanism for uplink and downlink QoS management. In particular, they are integral to the bandwidth allocation process. An SS requests uplink bandwidth on a per connection basis (implicitly identifying the service flow). Bandwidth is granted by the BS either as an aggregate of all grants for an SS (within a scheduling interval) or on a connection basis.

Connections, once established, may require active maintenance. The maintenance requirements vary depending upon the type of service connected. For example, unchannelized T1 services require virtually no connection maintenance since they have a constant bandwidth allocated every frame. Channelized T1 services require some maintenance due to the dynamic (but relatively slowly changing) bandwidth requirements if compressed, coupled with the requirement that full bandwidth be available on demand. IP services may require a substantial amount of ongoing maintenance due to their bursty nature and due to the high possibility of fragmentation. As with connection establishment, modifiable connections may require maintenance due to stimulus from either the SS or the network side of the connection.

Finally, connections may be terminated. This generally occurs only when a customer's service contract changes. The termination of a connection is stimulated by the BS or SS.

All three of these connection management functions are supported through the use of static configuration and dynamic addition, modification, and deletion of connections.

6.1 MAC service definition

This subclause defines the services between the MAC and the CSs. This is a logical interface. As such, the primitives described are informative. Their purpose is to describe the information that must necessarily be exchanged between the MAC and the CSs to enable each to perform its requirements as specified in the remainder of this document. This subclause does not impose message formats or state machines for the use of these primitives.

In a layered protocol system, the information flow across the boundaries between the layers can be defined in terms of primitives that represent different items of information and cause actions to take place. These primitives do not appear as such on the medium (the air interface) but serve to define more clearly the relations of the different layers. The semantics are expressed in the parameters that are conveyed with the primitives.

Since there are several sublayers, we divide the primitives into those providing MAC services to the CS above and a set of services provided to the external (non-IEEE Std 802.16-2001) layers above. The service access point (SAP) is shown in Figure 1.

6.1.1 Primitives

The IEEE Std 802.16-2001 MAC supports the following primitives at the MAC SAP:

MAC_CREATE_CONNECTION.request
MAC_CREATE_CONNECTION.indication
MAC_CREATE_CONNECTION.response
MAC_CREATE_CONNECTION.confirmation

MAC_CHANGE_CONNECTION.request
MAC_CHANGE_CONNECTION.indication
MAC_CHANGE_CONNECTION.response
MAC_CHANGE_CONNECTION.confirmation

MAC_TERMINATE_CONNECTION.request
MAC_TERMINATE_CONNECTION.indication
MAC_TERMINATE_CONNECTION.response
MAC_TERMINATE_CONNECTION.confirmation

MAC_DATA.request
MAC_DATA.indication

The use of these primitives to provide peer communication is shown in Figure 17. The initial request for service from a lower layer is provided by the "request" primitive. When this request is sent across the air link to the peer MAC sublayer, it generates an "indicate" primitive to inform the peer CS of the request; the convergence entity responds with a "response" to the MAC. Again this is sent across the air link to the MAC on the originating side, which sends a "confirm" primitive to the original requesting entity.

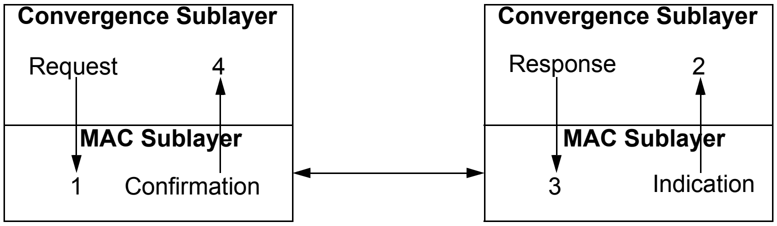


Figure 17—Use of primitives to request service of MAC sublayer and generate response

In some cases, it is not necessary to send information to the peer station and the "confirm" primitive is issued directly by the MAC on the originating side. Such cases may occur, for example, when the request is rejected by the MAC on the requesting side. In cases where it is necessary to keep the other side of the link informed, an unsolicited "response" may be sent, in turn leading to the generation of an unsolicited "confirmation" for benefit of the CS.

For actions other than DATA.request and DATA.indication, the initiating CS sends a REQUEST primitive to its MAC sublayer. The initiating side MAC sublayer sends the appropriate dynamic service request message (addition, change, or deletion; see 6.2.13.8) to the receiving MAC. The non-initiating side MAC sends an INDICATION primitive to its CS. The noninitiating CS responds with a RESPONSE primitive, stimulating its MAC to respond to the initiating side MAC with the appropriate Dynamic Service Response message. The initiating side MAC responds to its CS with a CONFIRMATION primitive and, if appropriate, with the appropriate dynamic service acknowledge message. At any point along the way, the request may be rejected (for lack of resources, etc.), terminating the protocol.

6.1.1.1 MAC_CREATE_CONNECTION.request

6.1.1.1.1 Function

This primitive is issued by a CS entity in a BS or SS unit to request the dynamic addition of a connection.

6.1.1.1.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_CREATE_CONNECTION.request
        (
        scheduling service type,
        convergence sublayer,
        service flow parameters,
        payload header suppression indicator,
        length indicator,
        encryption indicator,
        Packing on/off indicator,
        Fixed-length or variable-length SDU indicator,
        SDU length (only needed for fixed-length SDU connections),
        CRC request,
        ARQ parameters,
        sequence number
)
```

The scheduling service type (see 6.2.5) is one of the following: Unsolicited grant service (UGS), real-time polling service (rtPS), non-real-time polling service (nrtPS), and best effort (BE) service.

The CS parameter indicates which CS handles data received on this connection. If the value is zero, then no CS is used; other values for specific CSs are given in 11.4.9.

The service flow parameters include details on such issues as peak and average rate, or reference to a service flow. These parameters are the same as those in the dynamic service change request MAC management message.

The payload header suppression indicator specifies whether the SDUs on the service flow are to have their headers suppressed.

The packing on/off indicator specifies whether packing may be applied to the MAC SDUs on this connection. The fixed-length or variable-length SDU indicator specifies whether the SDUs on the service flow are fixed-length or variable-length.

The SDU length specifies the length of the SDU for a fixed-length SDU service flow.

The encryption indicator specifies that the data sent over this connection is to be encrypted, if ON. If OFF, then no encryption is used. The packing on/off indicator’s “on” value means that packing is allowed for the connection.

Cyclic redundancy check (CRC) request, if ON, requests that the MAC SDUs delivered over this connection are transported in MAC PDUs with a CRC added to them.

The automatic repeat request (ARQ) parameters are: whether or not ARQ is used for the connection, maximum retransmission limit, and acknowledgment window size. The sequence number is used to correlate this primitive with its response from the BS via the MAC.

6.1.1.1.3 When generated

This primitive is generated by a CS of a BS or SS unit to request the BS to set up a new connection.

6.1.1.1.4 Effect of receipt

If the primitive is generated on the SS side, the receipt of this primitive causes the MAC to pass the request (in the form of a dynamic service addition—request message) to the MAC entity in the BS. The SS MAC remembers the correlation between sequence number and the requesting convergence entity.

If the primitive is generated on the BS side, the BS checks the validity of the request and, if valid, chooses a CID and includes it in the Dynamic Service Addition—Request message (6.2.13.8.3) sent to the SS. This CID shall be returned to the requesting CS via the CONFIRM primitive. If the primitive originated at the SS, the actions of generating a CID and authenticating the request are deferred to the INDICATION/RESPONSE portion of the protocol.

6.1.1.2 MAC_CREATE_CONNECTION.indication

6.1.1.2.1 Function

This primitive is sent by the noninitiating MAC entity to the CS, to request the dynamic addition of a connection in response to the MAC sublayer receiving a Dynamic Service Addition—Request message. If the noninitiating MAC entity is at the base station, a CID is generated and the request is authenticated.

6.1.1.2.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_CREATE_CONNECTION.indication
        (
        service type,
        convergence sublayer,
        service flow parameters,
        sequence number
        )
```

Parameters: see MAC_CREATE_CONNECTION.request. The encryption and CRC flags are not delivered with the.indication primitive since they will have already been acted on by lower layers, to decrypt the data or to check a CRC, before the MAC SDU is passed up to the CS.

6.1.1.2.3 When generated

This primitive is generated by the MAC sublayer of the noninitiating side of the protocol when it receives a Dynamic Service Addition—Request message from the initiating side of the connection.

6.1.1.2.4 Effect of receipt

When the CS receives this primitive, it checks the validity of the request from the point of view of its own resources. It accepts or rejects the request via the MAC_CREATE_CONNECTION.response primitive.

If the connection request has originated on the SS side, the BS sends the CID to the SS side in this RESPONSE primitive. Otherwise, if the origin was the BS, the RESPONSE contains the CID contained in the DSA header bearing the indication.

6.1.1.3 MAC_CREATE_CONNECTION.response

6.1.1.3.1 Function

This primitive is issued by a noninitiating MAC entity in response to a MAC_CREATE_CONNECTION.indication requesting the creation of a new connection.

6.1.1.3.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_CREATE_CONNECTION.response
        (
        Connection ID,
        response code,
        response message,
        sequence number,
        ARQ parameters
        )
```

The Connection ID is returned to the requester for use with the traffic specified in the request. If the request is rejected, then this value shall be ignored.

The response code indicates success or the reason for rejecting the request.

The response message provides additional information to the requester, in type-length-value (TLV) format.

The sequence number is returned to the requesting entity to correlate this response with the original request.

The ARQ parameters are: whether or not ARQ is used for the connection, maximum retransmission limit and acknowledgment window size.

6.1.1.3.3 When generated

This primitive is generated by the noninitiating CS entity when it has received a MAC_CREATE_CONNECTION.indication.

6.1.1.3.4 Effect of receipt

The receipt of this primitive causes the MAC sublayer to send the Dynamic Service Addition—Response message to the requesting MAC entity. Once the Dynamic Service Addition—Acknowledgement is received, the MAC is prepared to pass data for this connection on to the air link.

6.1.1.4 MAC_CREATE_CONNECTION.confirmation

6.1.1.4.1 Function

This primitive confirms to a convergence entity that a requested connection has been provided. It informs the CS of the status of its request and provides a CID for the success case.

6.1.1.4.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_CREATE_CONNECTION.confirmation
        (
        Connection ID,
        response code,
        response message,
        sequence number
        )
```

Parameters: see MAC_CREATE_CONNECTION.response.

6.1.1.4.3 When generated

This primitive is generated by the initiating side MAC entity when it has received a Dynamic Service Addition—Response message.

6.1.1.4.4 Effect of receipt

The receipt of this primitive informs the convergence entity that the requested connection is available for transmission requests.

6.1.1.5 Changing an existing connection

Existing connections may be changed in their characteristics on a dynamic basis to, for example, reflect changing bandwidth requirements. The following primitives are used:

MAC_CHANGE_CONNECTION.request
MAC_CHANGE_CONNECTION.indication
MAC_CHANGE_CONNECTION.response
MAC_CHANGE_CONNECTION.confirmation

The semantics and effect of receipt of these primitives are the same as for the corresponding CREATE primitives, except that a new CID is not generated.

6.1.1.6 MAC_TERMINATE_CONNECTION.request

6.1.1.6.1 Function

This primitive is issued by a CS entity in a BS or SS unit to request the termination of a connection.

6.1.1.6.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_TERMINATE_CONNECTION.request
        (
        Connection ID
        )
```

The Connection ID parameter specifies which connection is to be terminated.

6.1.1.6.3 When generated

This primitive is generated by a CS of a BS or SS unit to request the termination of an existing connection.

6.1.1.6.4 Effect of receipt

If the primitive is generated on the SS side, the receipt of this primitive causes the MAC to pass the request to the MAC entity in the BS via the Dynamic Service Deletion—Request message. The base station checks the validity of the request, and if it is valid it terminates the connection.

If the primitive is generated on the base station side, it has already been validated and the base station MAC informs the SS by issuing a Dynamic Service Deletion—Request message.

6.1.1.7 MAC_TERMINATE_CONNECTION.indication

6.1.1.7.1 Function

This primitive is issued by a the MAC entity on the non-initiating side to request the termination of a connection in response to the receipt of a Dynamic Service Deletion—Request message.

6.1.1.7.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_TERMINATE_CONNECTION.indication
        (
        Connection ID
        )
```

The Connection ID parameter specifies which connection is to be terminated.

6.1.1.7.3 When generated

This primitive is generated by the MAC sublayer when it receives a Dynamic Service Deletion—Request message to terminate a connection, or when it finds it necessary for any reason to terminate a connection.

6.1.1.7.4 Effect of receipt

If the protocol was initiated by the SS, when it receives this primitive, the BS checks the validity of the request. In any case, the receiving CS returns the MAC_TERMINATE_CONNECTION.response primitive and deletes the CID from the appropriate polling and scheduling lists.

6.1.1.8 MAC_TERMINATE_CONNECTION.response

6.1.1.8.1 Function

This primitive is issued by a CS entity in response to a request for the termination of a connection.

6.1.1.8.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_TERMINATE_CONNECTION.response
        (
        Connection ID,
        response code,
        response message
        )
```

The Connection ID is returned to the requesting entity.

The response code indicates success or the reason for rejecting the request.

The response message provides additional information to the requester, in TLV format.

6.1.1.8.3 When generated

This primitive is generated by the CS entity when it has received a MAC_TERMINATE_CONNECTION.indication from its MAC sublayer.

6.1.1.8.4 Effect of receipt

The receipt of this primitive causes the MAC sublayer to pass the message to the initiating side via the Dynamic Service Deletion—Response message. The initiating MAC in turn passes the CONFIRM primitive to the requesting convergence entity. The convergence entity shall no longer use this CID for data transmission.

6.1.1.9 MAC_TERMINATE_CONNECTION.confirmation

6.1.1.9.1 Function

This primitive confirms to a convergence entity that a requested connection has been terminated.

6.1.1.9.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_TERMINATE_CONNECTION.confirmation
        (
        Connection ID,
        response code,
        response message
        )
```

Parameters: see MAC_TERMINATE_CONNECTION.response.

6.1.1.9.3 When generated

This primitive is generated by the MAC entity when it has received a Dynamic Service Deletion—Response message.

6.1.1.9.4 Effect of receipt

The receipt of this primitive informs the convergence entity that a connection has been terminated. The convergence entity shall no longer use this CID for data transmission.

6.1.1.10 MAC_DATA.request

6.1.1.10.1 Function

This primitive defines the transfer of data to the MAC entity from a CS SAP.

6.1.1.10.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_DATA.request
        (
        Connection ID,
        length,
        data,
        discard-eligible flag,
        encryption flag
        )
```

The Connection ID parameter specifies the connection over which the data is to be sent; the service class is implicit in the Connection ID.

The length parameter specifies the length of the MAC SDU in bytes.

The data parameter specifies the MAC SDU as received by the local MAC entity.

The discard-eligible flag specifies whether the MAC SDU is to be preferentially discarded in the event of link congestion and consequent buffer overflow.

The encryption flag specifies that the data sent over this connection is to be encrypted, if ON. If OFF, then no encryption is used.

6.1.1.10.3 When generated

This primitive is generated by a CS whenever a MAC SDU is to be transferred to a peer entity or entities.

6.1.1.10.4 Effect of receipt

The receipt of this primitive causes the MAC entity to process the MAC SDU through the MAC sublayer and pass the appropriately formatted PDUs to the PHY transmission CS for transfer to peer MAC sublayer entities, using the CID specified.

6.1.1.11 MAC_DATA.indication

6.1.1.11.1 Function

This primitive defines the transfer of data from the MAC to the CS. The specific CS to receive the indicate message is implicit in the CID.

6.1.1.11.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MAC_DATA.indication
        (
        Connection ID,
        length,
        data,
        reception status,
        CS pass through,
        encryption flag
        )
```

The Connection ID parameter specifies the connection over which the data was sent.

The length parameter specifies the length of the data unit in bytes.

The data parameter specifies the MAC SDU as received by the local MAC entity.

The reception status parameter indicates transmission success or failure for those PDUs received via the MAC_DATA.indication.

6.1.1.11.3 When generated

This primitive is generated whenever an MAC SDU is to be transferred to a peer convergence entity or entities.

6.1.1.11.4 Effect of receipt

The effect of receipt of this primitive by a convergence entity is dependent on the validity and content of the MAC SDU. The choice of CS is determined by the CID over which the MAC SDU was sent.

6.1.2 MAC service stimulation of dynamic service messages

This subclause describes the logical interaction between the MAC Service primitives and the dynamic service addition, change, or deletion (DSx) messages.

The sequence of logical MAC SAP events and the associated actual MAC events effecting a CS-stimulated connection creation are shown in Figure 18.

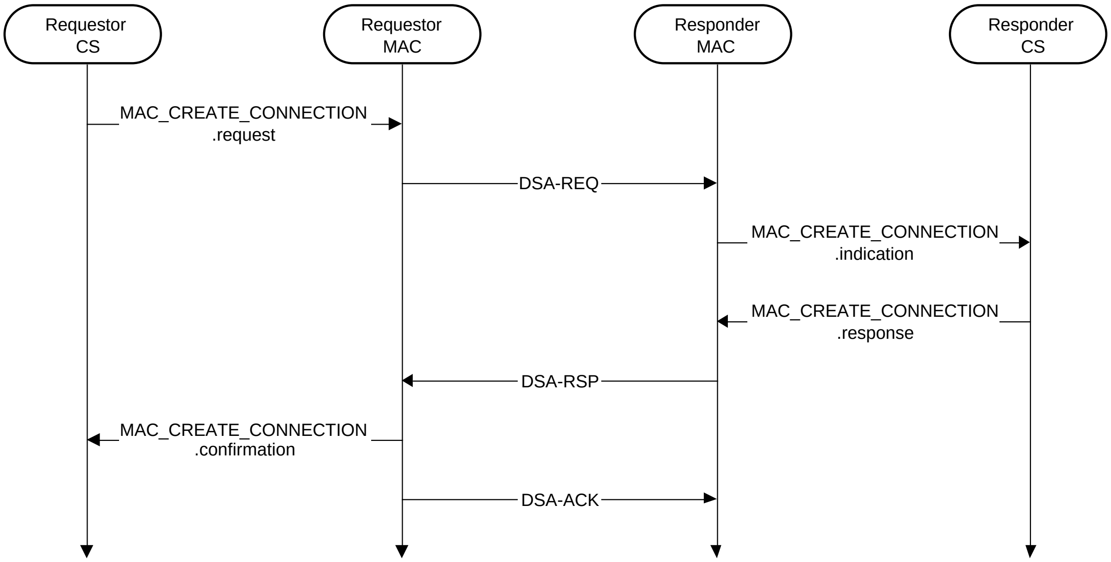

Requestor CS
Requestor MAC
Responder MAC
Responder CS
MAC_CREATE_CONNECTION .request
DSA-REQ
MAC_CREATE_CONNECTION .indication
MAC_CREATE_CONNECTION .response
DSA-RSP
MAC_CREATE_CONNECTION .confirmation
DSA-ACK

Figure 18—MAC SAP event and MAC event sequence for connection creation stimulated by convergence sublayer

The sequence of logical MAC SAP events and the associated actual MAC events effecting a CS stimulated connection change are shown in Figure 19.

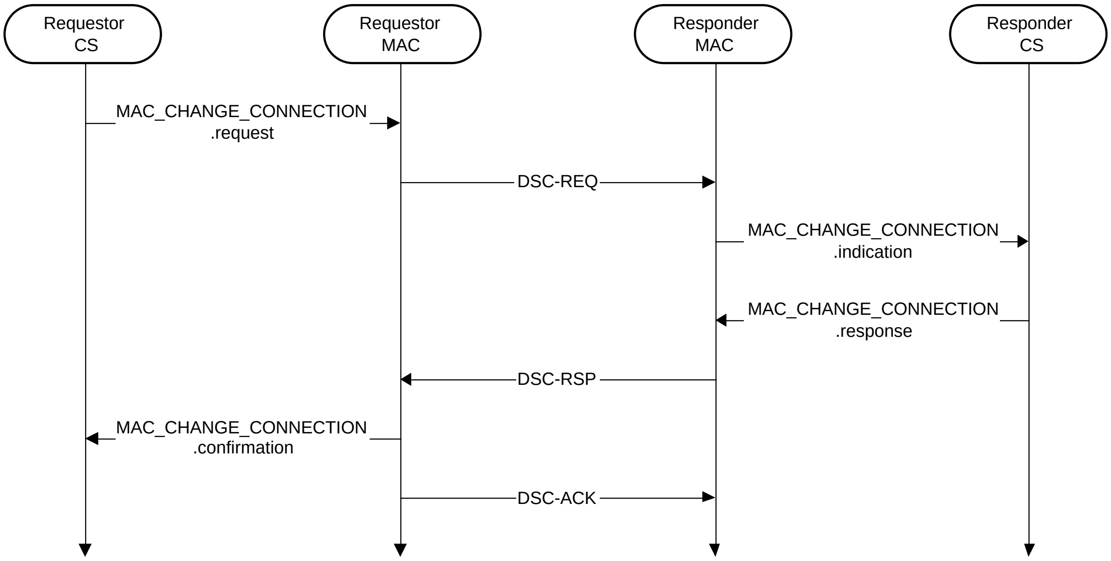

Requestor CS
Requestor MAC
Responder MAC
Responder CS
MAC_CHANGE_CONNECTION .request
DSC-REQ
MAC_CHANGE_CONNECTION .indication
MAC_CHANGE_CONNECTION .response
DSC-RSP
MAC_CHANGE_CONNECTION .confirmation
DSC-ACK

Figure 19—MAC SAP event and MAC event sequence for connection change stimulated by convergence sublayer

The sequence of logical MAC SAP events and the associated actual MAC events effecting a CS stimulated connection deletion are shown in Figure 20.

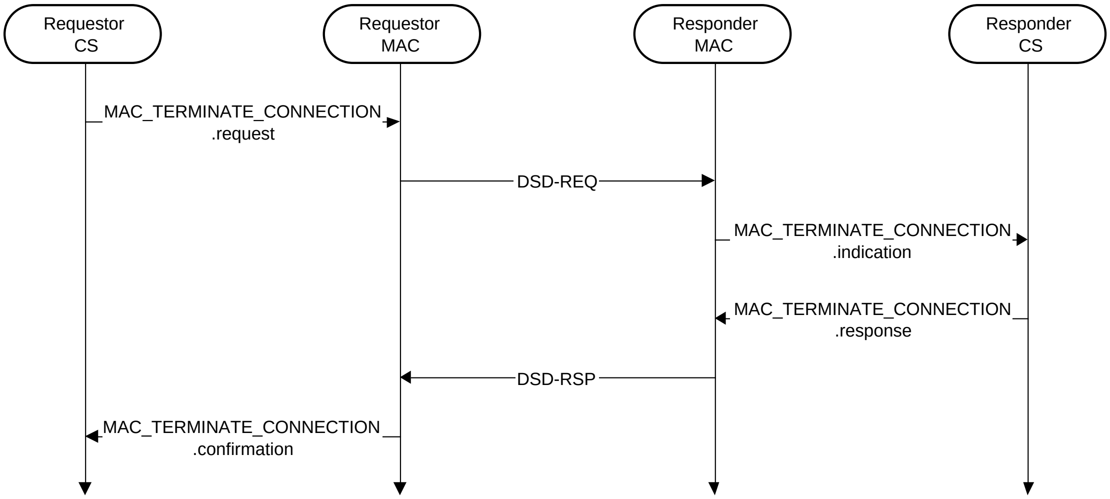

Requestor CS
Requestor MAC
Responder MAC
Responder CS
MAC_TERMINATE_CONNECTION .request
DSD-REQ
MAC_TERMINATE_CONNECTION .indication
MAC_TERMINATE_CONNECTION .response
DSD-RSP
MAC_TERMINATE_CONNECTION .confirmation

Figure 20—MAC SAP event and MAC event sequence for connection deletion stimulated by convergence sublayer

6.2 Data/Control plane

6.2.1 Addressing and connections

Each SS shall have a 48-bit universal MAC address, as defined in IEEE Std 802®-2001. This address uniquely defines the SS from within the set of all possible vendors and equipment types. It is used during the registration process to establish the appropriate connections for an SS. It is also used as part of the authentication process by which the BS and SS each verify the identity of each other.

Connections are identified by a 16-bit CID. At SS initialization, three management connections in each direction (uplink and downlink) shall be established between the SS and the BS. These CIDs shall be assigned in the RNG-RSP and REG-RSP messages and reflect the fact that there are inherently three different QoS of management traffic between an SS and the BS. The basic connection is used by the BS MAC and SS MAC to exchange short, time-urgent MAC management messages. The primary management connection is used by the BS MAC and SS MAC to exchange longer, more delay tolerant MAC management messages. Table 13 specifies which MAC Management Messages are transferred on which of these two connections. Finally, the Secondary Management Connection is used by the BS and SS to transfer delay tolerant, standards based [Dynamic Host Configuration Protocol (DHCP), Trivial File Transfer Protocol (TFTP), SNMP, etc.) management messages. These messages are carried in IP packets, as specified in 5.2.7. The packets shall be assigned to the secondary management connection based on the full IP source address in the uplink and the full IP destination address in the downlink.

For bearer services, the higher layers of the BS set up connections based upon the provisioning information distributed to the BS. The registration of an SS, or the modification of the services contracted at an SS, stimulates the higher layers of the BS to initiate the setup of the connections.

The CID can be considered a connection identifier even for nominally connectionless traffic like IP, since it serves as a pointer to destination and context information. The use of a 16-bit CID permits a total of 64K connections within each downlink and uplink channel.

Requests for transmission are based on these CIDs, since the allowable bandwidth may differ for different connections, even within the same service type. For example, an SS unit serving multiple tenants in an office building would make requests on behalf of all of them, though the contractual service limits and other connection parameters may be different for each of them.

Many higher-layer sessions may operate over the same wireless CID. For example, many users within a company may be communicating with TCP/IP to different destinations, but since they all operate within the same overall service parameters, all of their traffic is pooled for request/grant purposes. Since the original LAN source and destination addresses are encapsulated in the payload portion of the transmission, there is no problem in identifying different user sessions.

The type of service is implicit in the CID; it is accessed by a lookup indexed by the CID.

6.2.2 MAC PDU formats

MAC Protocol Data Units (MAC PDUs) shall be of the form illustrated in Figure 21. Each PDU shall begin with a fixed-length Generic MAC Header. The header may be followed by the Payload of the MAC PDU. If present, the Payload shall consist of zero or more subheaders and zero or more MAC SDUs and/or fragments thereof. The payload information may vary in length, so that a MAC PDU may represent a variable number of bytes. This allows the MAC to tunnel various higher layer traffic types without knowledge of the formats or bit patterns of those messages.

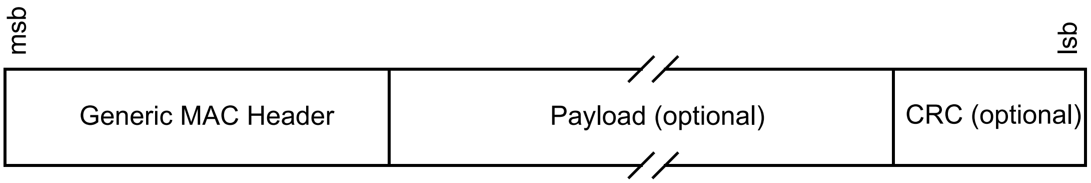


Figure 21—MAC PDU formats

A MAC PDU may contain a CRC, as described in 6.2.3.5.

6.2.2.1 MAC header formats

Two MAC header formats are defined. The first is the Generic MAC Header that begins each MAC PDU containing either MAC management messages or CS data. The second is the Bandwidth Request Header used to request additional bandwidth. The single-bit Header Type (HT) field distinguishes the Generic and Bandwidth Request Header formats. The HT field shall be set to zero for the Generic Header and to one for a Bandwidth Request Header.

The MAC header formats are defined in Table 2.

Table 2—MAC header format

Syntax	Size	Notes
MAC Header() {		
HT	1 bit	0 = Generic MAC Header 1 = Bandwidth Request Header
EC	1 bit	If HT = 1, EC = 0
if (HT == 0) {		
Type	6 bits	
Rsv	1 bit	Reserved; set to 0
CI	1 bit	
EKS	2 bits	
Rsv	1 bit	Reserved; set to 0
LEN	11 bits	
}		
else {		
Type	6 bits	
BR	16 bits	
}		
CID	16 bits	
HCS	8 bits	
}		

6.2.2.1.1 Generic MAC header

The Generic MAC header is illustrated in Figure 22.

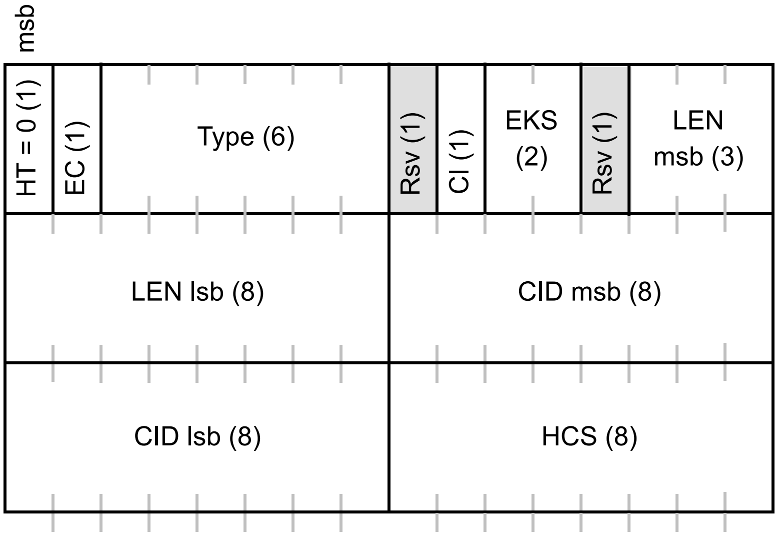


Figure 22—Generic MAC header format

The fields of the Generic MAC header are defined in Table 3. Every header is encoded, starting with the HT and encryption control (EC) fields. The coding of these fields is such that the first byte of a MAC header shall never have the value of 0xFX. This prevents false detection on the stuff byte used in the transmission CS.

Table 3—Generic MAC header fields

Name	Length (bits)	Description
CI	1	CRC Indicator 1 = CRC is appended to the PDU 0 = No CRC is appended
CID	16	Connection Identifier
EC	1	Encryption Control 0 = Payload is not encrypted 1 = Payload is encrypted
EKS	2	Encryption Key Sequence The index of the Traffic Encryption Key and Initialization Vector used to encrypt the payload. This field is only meaningful if the Encryption Control field is set to 1.
HCS	8	Header Check Sequence An 8-bit field used to detect errors in the header. The generator polynomial is $g(D)=D^8 + D^2 + D + 1$.
HT	1	Header Type. Shall be set to zero.
LEN	11	Length The length in bytes of the MAC PDU including the MAC header.
Type	6	This field indicates the payload type, including presence of subheaders.

The allowed values for the Type field are listed in Table 4 and Table 5.

Table 4—Downlink type encoding

Type	Description
0x00	No subheaders present
0x01	*reserved*
0x02	Packing subheader present
0x03	*reserved*
0x04	Fragmentation subheader present
0x05-0x3F	*reserved*

Table 5—Uplink type encoding

Type	Description
0x00	No subheaders present
0x01	Grant Management subheader present
0x02	Packing subheader present
0x03	Both Grant Management and Packing subheaders present
0x04	Fragmentation subheader present
0x05	Fragmentation and grant management subheaders present
0x06- 0x3F	*reserved*

6.2.2.1.2 Bandwidth Request Header

The Bandwidth Request PDU shall consist of Bandwidth Request Header alone and shall not contain a payload. The Bandwidth Request Header is illustrated in Figure 23.

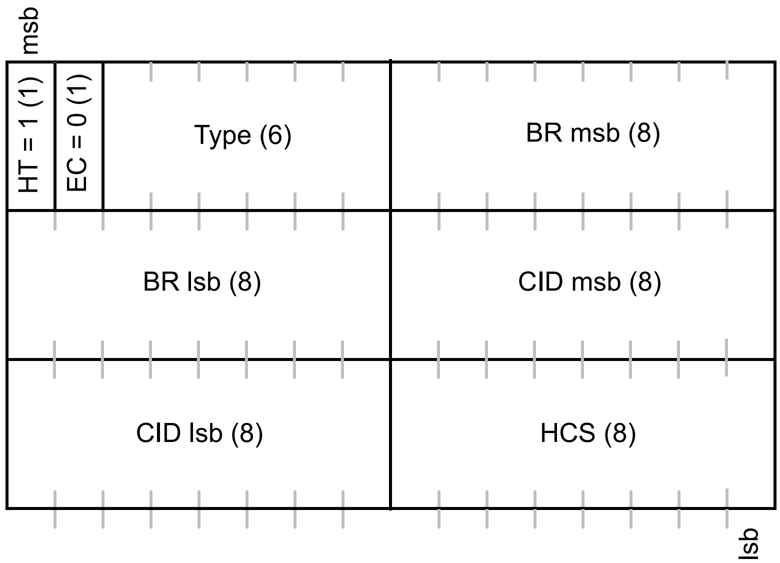


Figure 23—Bandwidth request header format

The Bandwidth Request shall have the following properties:

a) The length of the header shall always be 6 bytes.

b) The EC field shall be set to 0, indicating no encryption.

c) The CID shall indicate the service flow for which uplink bandwidth is requested.

d) The Bandwidth Request (BR) field shall indicate the number of bytes requested.

e) The allowed types for bandwidth requests are “000000” for incremental and “000001” for aggregate.

An SS receiving a Bandwidth Request Header on the downlink shall discard the PDU.

The fields of the Bandwidth Request Header are defined in Table 6. Every header is encoded, starting with the HT and EC fields. The coding of these fields is such that the first byte of a MAC header shall never have the value of 0xFX. This prevents false detection on the stuff byte used in the transmission CS.

Table 6—Bandwidth request header fields

Name	Length (bits)	Description
BR	16	Bandwidth Request The number of bytes of uplink bandwidth requested by the SS. The bandwidth request is for the CID. The request shall not include any PHY overhead.
CID	16	Connection Identifier
EC	1	Always set to zero.
HCS	8	Header Check Sequence An 8-bit field used to detect errors in the header. The generator polynomial is $g(D)=D^8 + D^2 + D + 1$.
HT	1	Header Type = 1
Type	6	Indicates the type of bandwidth request header

6.2.2.2 MAC subheaders

Three types of MAC subheaders may be present. The per-PDU subheaders (the Fragmentation subheader and the Grant Management subheaders) may be inserted in MAC PDUs immediately following the Generic MAC Header if so indicated by the Type field. If both the Fragmentation and Grant Management subheaders are indicated, the Grant Management subheader shall come first.

The only per-SDU subheader is the Packing subheader. It may be inserted before each MAC SDU if so indicated by the Type field. The Packing and Fragmentation subheaders are mutually exclusive and shall not both be present within the same MAC PDU.

If present, all per-PDU subheaders shall always come before the first per-SDU subheader.

6.2.2.2.1 Fragmentation subheader

The Fragmentation subheader (FSH) is shown in Table 7.

Table 7—Fragmentation subheader format

Syntax	Size	Notes
Fragmentation subheader () {		
FC	2 bits	
FSN	3 bits	
reserved for CS use	3 bits	
}		

The fields of the Fragmentation subheader are defined in Table 8.

Table 8—Fragmentation subheader fields

Name	Length (bits)	Description
FC	2	Fragmentation Control Indicates the fragmentation state of the payload: 00 = no fragmentation 01 = last fragment 10 = first fragment 11 = continuing (middle) fragment
FSN	3	Fragmentation Sequence Number Defines the sequence number of the current SDU fragment. This field increments by one (modulo 8) for each fragment, including unfragmented SDUs.

6.2.2.2.2 Grant Management subheader

The Grant Management (GM) subheader is two bytes in length and is used by the SS to convey bandwidth management needs to the BS. This subheader is encoded differently based upon the type of uplink scheduling service for the connection (as given by the CID). The use of this subheader is defined in 6.2.6. The Grant Management subheader is shown in Table 9. Its fields are defined in Table 10.

Table 9—Grant Management subheader format

Syntax	Size	Notes
Grant Management subheader() {		
if (scheduling service type == UGS) {		
SI	1 bit	
PM	1 bit	
reserved	14 bits	Set to 0
}		
else {		
PiggyBack Request	16 bits	
}		
}		

Table 10—Grant Management subheader fields

Name	Length (bits)	Description
PBR	16	PiggyBack Request The number of bytes of uplink bandwidth requested by the SS. The bandwidth request is for the CID. The request shall not include any PHY overhead.
PM	1	Poll-Me 0 = No action 1 = Used by the SS to request a bandwidth poll.
SI	1	Slip Indicator 0 = No action 1 = Used by the SS to indicate a slip of uplink grants relative to the uplink queue depth.

6.2.2.2.3 Packing subheader

When Packing (see 6.2.3.4) is used, the MAC may pack multiple SDUs into a single MAC PDU. When packing variable-length MAC SDUs, the MAC precedes each one with a Packing subheader. The Packing subheader is defined in Table 11.

Table 11—Packing subheader format

Syntax	Size	Notes
Packing sub-header () {		
FC	2 bits	
FSN	3 bits	
Length	11 bits	
}		

The fields of the packing subheader are defined in Table 12.

Table 12—Packing subheader fields

Name	Length (bits)	Description
FC	2	Fragmentation Control Indicates the fragmentation state of the payload: 00 = no fragmentation 01 = last fragment 10 = first fragment 11 = continuing (middle) fragment
FSN	3	Fragmentation Sequence Number Defines the sequence number of the current SDU fragment. This field increments by one (modulo 8) for each fragment, including unfragmented SDUs.
Length	11	The length in bytes of the MAC SDU or SDU fragment, including the two-byte packing subheader.

6.2.2.3 MAC Management Messages

A set of MAC Management Messages are defined. These messages shall be carried in the Payload of the MAC PDU. All MAC Management Messages begin with a Management Message Type field and may contain additional fields. MAC Management Messages on the Basic, Broadcast, and Initial Ranging connections shall neither be fragmented nor packed. MAC Management Messages on the Primary Management Connection may be packed and/or fragmented. The format of the Management Message is given in Figure 24. The encoding of the Management Message Type field is given in Table 13. MAC management messages shall not be carried on Transport Connections.

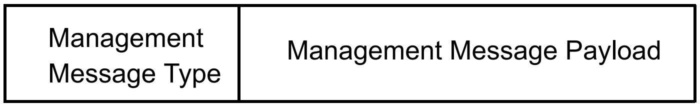


Figure 24—MAC Management Message format

Table 13—MAC Management Messages

Type	Message name	Message description	Connection
0	UCD	Uplink Channel Descriptor	Broadcast
1	DCD	Downlink Channel Descriptor	Broadcast
2	DL-MAP	Downlink Access Definition	Broadcast
3	UL-MAP	Uplink Access Definition	Broadcast
4	RNG-REQ	Ranging Request	Initial Ranging or Basic
5	RNG-RSP	Ranging Response	Initial Ranging or Basic
6	REG-REQ	Registration Request	Primary Management
7	REG-RSP	Registration Response	Primary Management
8	*reserved*		
9	PKM-REQ	Privacy Key Management Request	Primary Management
10	PKM-RSP	Privacy Key Management Response	Primary Management
11	DSA-REQ	Dynamic Service Addition Request	Primary Management
12	DSA-RSP	Dynamic Service Addition Response	Primary Management
13	DSA-ACK	Dynamic Service Addition Acknowledge	Primary Management
14	DSC-REQ	Dynamic Service Change Request	Primary Management
15	DSC-RSP	Dynamic Service Change Response	Primary Management
16	DSC-ACK	Dynamic Service Change Acknowledge	Primary Management
17	DSD-REQ	Dynamic Service Deletion Request	Primary Management
18	DSD-RSP	Dynamic Service Deletion Response	Primary Management
19		*reserved for future use*	
20		*reserved for future use*	
21	MCA-REQ	Multicast Assignment Request	Basic
22	MCA-RSP	Multicast Assignment Response	Basic
23	DBPC-REQ	Downlink Burst Profile Change Request	Basic
24	DBPC-RSP	Downlink Burst Profile Change Response	Basic
25	RES-CMD	Reset Command	Basic
26	SBC-REQ	SS Basic Capability Request	Basic

Table 13—MAC Management Messages *(continued)*

Type	Message name	Message description	Connection
27	SBC-RSP	SS Basic Capability Response	Basic
28	CLK-CMP	SS network clock comparison	Broadcast
29	DREG-CMD	De/Re-register Command	Basic
30	DSX-RVD	DSx Received Message	Primary Management
31	TFTP-CPLT	Config File TFTP Complete Message	Primary Management
32	TFTP-RSP	Config File TFTP Complete Response	Primary Management
33-255		*reserved for future use*	

6.2.2.3.1 Downlink Channel Descriptor (DCD) message

A DCD shall be transmitted by the BS at a periodic interval (Table 118) to define the characteristics of a downlink physical channel.

The message parameters following the channel ID and configuration change count shall be encoded in a TLV form in which the type and length fields are each 1 byte long.

Table 14—DCD message format

Syntax	Size	Notes
DCD_Message_Format() {		
Management Message Type = 1	8 bits	
Downlink channel ID	8 bits	
Configuration Change Count	8 bits	
TLV Encoded information for the overall channel	Variable	TLV specific
Begin PHY Specific Section {		See applicable PHY section
for (*i* = 1; *i* <= *n*; *i*++) {		For each downlink burst profile 1 to *n*
Downlink_Burst_Profile		PHY specific
}		
}		
}		

A BS shall generate DCDs in the format shown in Table 14, including all of the following parameters:

Configuration Change Count

Incremented by one (modulo 256) by the BS whenever any of the values of this channel descriptor change. If the value of this count in a subsequent DCD remains the same, the SS can quickly decide that the remaining fields have not changed and may be able to disregard the remainder of the message.

Downlink Channel ID

The identifier of the downlink channel to which this Message refers. This identifier is arbitrarily chosen by the BS and is unique only within the MAC-Sublayer domain. This acts as a local identifier for transactions such as ranging.

The message parameters following the Configuration Change Count shall be encoded in a TLV form (see 11.1.2). All channel encodings (see 11.1.2.1) shall appear first before the Downlink_Burst_Profile encodings.

The Downlink_Burst_Profile is a compound TLV encoding that defines, and associates with a particular Downlink Interval Usage Code (DIUC), the PHY characteristics that shall be used with that DIUC. Within each Downlink_Burst_Profile shall be an unordered list of PHY attributes, encoded as TLV values (see 11.1.2.2). Each interval is assigned a DIUC by the DL-MAP message. A Downlink_Burst_Profile shall be included for each DIUC to be used in the DL-MAP unless the PHY's Downlink_Burst_Profile is explicitly known.

Downlink_Burst_Profile contents are defined separately for each PHY specification in Clause 8.

6.2.2.3.2 Downlink Map (DL-MAP) message

The DL-MAP message defines the access to the downlink information. If the length of the DL-MAP message is a nonintegral number of bytes, the LEN field in the MAC header is rounded up to the next integral number of bytes. The message shall be padded to match this length, but the SS shall disregard the 4 pad bits.

A BS shall generate DL-MAP messages in the format shown in Table 15, including all of the following parameters:

PHY Synchronization

The PHY synchronization field is dependent on the PHY specification used. The encoding of this field is given in each PHY specification separately.

DCD Count

Matches the value of the configuration change count of the DCD, which describes the down burst profiles that apply to this map.

Base Station ID

The Base Station ID is a 48-bit long field identifying the BS. The Base Station ID shall be programmable. The most significant 24 bits shall be used as the operator ID. This is a network management hook that can be combined with the Downlink Channel ID of the DCD message for handling edge-of-sector and edge-of-cell situations.

Number of Elements

The number of information elements that follows.

The encoding of the remaining portions of the DL-MAP message is PHY-specification dependent and may be absent. Refer to the appropriate PHY specification.

Table 15—DL-MAP message format

Syntax	Size	Notes
DL-MAP_Message_Format() {		
Management Message Type = 2	8 bits	
PHY Synchronization Field	Variable	See appropriate PHY specification.
DCD Count	8 bits	
Base Station ID	48 bits	
Number of DL-MAP Elements *n*	16 bits	
Begin PHY Specific Section {		See applicable PHY section.
for (*i* = 1; *i* <= *n*; *i*++) {		For each DL-MAP element 1 to *n*.
DL_MAP_Information_Element()	Variable	See corresponding PHY specification.
if !(byte boundary) {		
Padding Nibble	4 bits	Padding to reach byte boundary.
}		
}		
}		
}		

6.2.2.3.3 Uplink Channel Descriptor message

An Uplink Channel Descriptor (UCD) shall be transmitted by the BS at a periodic interval (Table 118) to define the characteristics of an uplink physical channel. A separate UCD Message shall be transmitted for each active uplink channel associated with the downlink channel.

A BS shall generate UCDs in the format shown in Table 16, including all of the following parameters:

Configuration Change Count
Incremented by one (modulo 256) by the BS whenever any of the values of this channel descriptor change. If the value of this count in a subsequent UCD remains the same, the SS can quickly decide that the remaining fields have not changed and may be able to disregard the remainder of the message. This value is also referenced from the UL-MAP messages.

Mini-slot Size
The size n of the minislot for this uplink channel in units of Physical Slots. Allowable values are $n = 2^m$, where m is an integer ranging from 0 through 7.

Uplink Channel ID
The identifier of the uplink channel to which this Message refers. This identifier is arbitrarily chosen by the BS and is only unique within the MAC-Sublayer domain.

Ranging Backoff Start
Initial backoff window size for initial ranging contention, expressed as a power of 2. Values of *n* range 0–15 (the highest order bits shall be unused and set to 0).

Ranging Backoff End
Final backoff window size for initial ranging contention, expressed as a power of 2. Values of *n* range 0–15 (the highest order bits shall be unused and set to 0).

Request Backoff Start
Initial backoff window size for contention data and requests, expressed as a power of 2. Values of *n* range 0–15 (the highest order bits shall be unused and set to 0).

Request Backoff End
Final backoff window size for contention requests, expressed as a power of 2. Values of *n* range 0–15 (the highest order bits shall be unused and set to 0).

Table 16—UCD message format

Syntax	Size	Notes
UCD_Message_Format() {		
Management Message Type = 0	8 bits	
Uplink channel ID	8 bits	
Configuration Change Count	8 bits	
Minislot size	8 bits	
Ranging Backoff Start	8 bits	
Ranging Backoff End	8 bits	
Request Backoff Start	8 bits	
Request Backoff End	8 bits	
TLV Encoded information for the overall channel	Variable	TLV specific
Begin PHY Specific Section {		See applicable PHY section.
for (*i* = 1; *i* <= *n*; *i*++) {		For each uplink burst profile 1 to *n*.
Uplink_Burst_Profile	Variable	PHY specific
}		
}		
}		

To provide for flexibility, the remaining message parameters shall be encoded in a TLV form (see 11.1.1). All Channel encodings (see 11.1.1.1) shall appear first before the Uplink_Burst_Profile encodings.

The Uplink_Burst_Profile is a compound TLV encoding that defines, and associates with a particular UIUC, the PHY characteristics that shall be used with that UIUC. Within each Uplink_Burst_Profile shall be an unordered list of PHY attributes, encoded as TLV values (see 11.1.1.2 for an example applicable to the 10–66 GHz PHY specification). Each interval is assigned a UIUC by the UL-MAP message. An Uplink_Burst_Profile shall be included for each UIUC to be used in the UL-MAP.

Uplink_Burst_Profile contents are defined separately for each PHY specification in Clause 8.

6.2.2.3.4 Uplink Map (UL-MAP) message

The UL-MAP message allocates access to the uplink channel. The UL-MAP message shall be as shown in Table 17.

Table 17—UL-MAP message format

Syntax	Size	Notes
UL-MAP_Message_Format() {		
Management Message Type = 3	8 bits	
Uplink Channel ID	8 bits	
UCD Count	8 bits	
Number of UL-MAP Elements *n*	16 bits	
Allocation Start Time	32 bits	
Begin PHY Specific Section {		See applicable PHY section.
for (*i* = 1; *i* <= *n*; *i*++) {		For each UL-MAP element 1 to *n*.
UL_MAP_Information_Element()	Variable	See corresponding PHY specification.
}		
}		
}		

The BS shall generate the UL-MAP with the following parameters:

Uplink Channel ID
The identifier of the uplink channel to which this Message refers.

UCD Count
Matches the value of the Configuration Change Count of the UCD which describes the uplink burst profiles which apply to this map.

Number of Elements
Number of information elements in the map.

Alloc Start Time
Effective start time of the uplink allocation defined by the UL-MAP in units of minislots.

Map Information Elements
Each Information Element (IE) consists of at least three of the following fields:

1) Connection Identifier (CID)
2) Uplink Interval Usage Code (UIUC)
3) Offset

IEs define uplink bandwidth allocations. Each UL-MAP message shall contain at least one IE that marks the end of the last allocated burst. The IEs shall be in strict chronological order within the UL-MAP.

The CID represents the assignment of the IE to either a unicast, multicast, or broadcast address. When specifically addressed to allocate a bandwidth grant, the CID shall be either the Basic CID of the SS or a

Transport CID for one of the connections of the SS. An UIUC shall be used to define the type of uplink access and the uplink burst profile associated with that access. An Uplink_Burst_Profile shall be included in the UCD for each UIUC to be used in the UL-MAP.

6.2.2.3.5 Ranging Request (RNG-REQ) message

An RNG-REQ shall be transmitted by the SS at initialization and periodically on the request of the BS to determine network delay and to request power and/or downlink burst profile change. The format of the RNG-REQ message is shown in Table 18. The RNG-REQ message may be sent in Initial Maintenance and Station Maintenance intervals.

Table 18—RNG-REQ message format

Syntax	Size	Notes
RNG-REQ_Message_Format() {		
Management Message Type = 4	8 bits	
Downlink Channel ID	8 bits	
Pending Until Complete	8 bits	
TLV Encoded Information	Variable	TLV specific
}		

The CID field in the MAC header shall assume the following values when sent in an Initial Maintenance interval:

a) Initial ranging CID if SS is attempting to join the network.

b) Initial ranging CID if SS has not yet registered and is changing downlink (or both downlink and uplink) channels as directed by the downloaded SS Configuration File (9.2).

c) Basic CID (previously assigned in RNG-RSP) if SS has not yet registered and is changing uplink channel as directed by the downloaded SS Configuration File (9.2).

d) Basic CID (previously assigned in RNG-RSP) if SS is registered and is changing uplink channel.

e) In all other cases, the Basic CID is used as soon as one is assigned in the RNG-RSP message.

If sent in a Station Maintenance interval, the CID is always equal to the Basic CID.

The parameters described below are present in the RNG-REQ message. Note that the length of the RNG-REQ message sent in an Initial Maintenance Interval is fixed, containing all of its TLVs.

Downlink Channel ID
The identifier of the downlink channel on which the SS received the UCD describing the uplink on which this ranging request message is to be transmitted. This is an 8-bit field.

Pending Until Complete
If zero, then all previous Ranging Response attributes have been applied prior to transmitting this request. If nonzero, then this is time estimated to be needed to complete assimilation of ranging parameters. Units are in unsigned centiseconds (10 ms).

All other parameters are coded as TLV tuples as defined in 11.1.3.

The following parameters shall be included in the RNG-REQ message:

Requested Downlink Burst Profile
SS MAC Address
Ranging Anomalies

6.2.2.3.6 Ranging response (RNG-RSP) message

A RNG-RSP shall be transmitted by the BS in response to received RNG-REQ or to send corrections based on measurements that have been done on other received data or MAC messages. From the point of view of the SS, reception of a RNG-RSP is stateless. In particular, the SS shall be prepared to receive an RNG-RSP at any time, not just following a RNG-REQ.

The initial RNG-RSP Message should be transmitted using a downlink burst profile that is sufficiently robust to provide for adequate reception.

To provide for flexibility, the Message parameters following the Uplink Channel ID shall be encoded in a TLV form.

A BS shall generate RNG-RSPs in the form shown in Table 19, including all of the following parameters:

Uplink Channel ID
The identifier of the uplink channel on which the BS received the RNG-REQ to which this response refers. This is an 8-bit quantity.

All other parameters are coded as TLV tuples, as defined in 11.1.4.

Table 19—RNG-RSP message format

Syntax	Size	Notes
RNG-RSP_Message_Format() {		
Management Message Type = 5	8 bits	
Uplink Channel ID	8 bits	
TLV Encoded Information	Variable	TLV specific
}		

The following parameters shall be included in the RNG-RSP message:

Timing Adjust Information
Power Adjust Information
Frequency Adjust Information
Ranging Status

The following parameters may be included in the RNG-RSP message:

Downlink Frequency Override
Uplink Channel ID Override
Downlink Operational Burst Profile
Basic CID
A required parameter if the RNG-RSP message is being sent on the Initial Ranging CID in response to a RNG-REQ message that was sent on the Initial Ranging CID.
Primary Management CID
A required parameter if the RNG-RSP message is being sent on the Initial Ranging CID in response to a RNG-REQ message that was sent on the Initial Ranging CID.
SS MAC Address (48-bit)
A required parameter when the CID in the MAC header is the Initial Ranging CID.

6.2.2.3.7 Registration Request (REG-REQ) message

An REG-REQ shall be transmitted by an SS at initialization. An SS shall generate REG-REQs in the form shown in Table 20.

Table 20—REG-REQ message format

Syntax	Size	Notes
REG-REQ_Message_Format() {		
Management Message Type = 6	8 bits	
TLV Encoded Information	Variable	TLV Specific
}		

An SS shall generate REG-REQs including the following parameters:

Primary Management CID *(in the Generic MAC Header)*
The CID in the Generic MAC Header is the Primary Management CID for this SS, as assigned in the RNG-RSP message.

All other parameters are coded as TLV tuples.

The REG-REQ shall contain the following TLVs:

HMAC Tuple (see 11.4.10)
Uplink (UL) CID Support (see 11.4.1.1)

The REG-REQs may contain the following TLV parameters stored in or generated by the SS:

Vendor ID Encoding (of the SS; see 11.4.3)
SS Capabilities Encodings (excluding UL CID support, physical parameters supported, and bandwidth allocation support) (see 11.4.1)

6.2.2.3.8 Registration Response (REG-RSP) message

A REG-RSP shall be transmitted by the BS in response to received REG-REQ.

To provide for flexibility, the message parameters following the response field shall be encoded in a TLV format.

Table 21—REG-RSP message format

Syntax	Size	Notes
REG-RSP_Message_Format() {		
Management Message Type = 7	8 bits	
Response	8 bits	
TLV Encoded Information	Variable	TLV specific
}		

A BS shall generate REG-RSPs in the form shown in Table 21, including both of the following parameters:

CID ***(in the Generic MAC Header)***
The CID in the Generic MAC Header is the Primary Management CID for this SS.
Response
A one-byte quantity with one of the two values:
0 = OK
1 = Message authentication failure

The following parameters shall be included in the REG-RSP:

MAC Version (see 11.4.4)
Secondary Management CID
Hashed message authentication code (HMAC) Tuple

The following parameter shall be included in the REG-RSP if found in the REG-REQ or if the BS requires the use of a non-default value:

SS Capabilities (see 11.4.1)
The BS response to the capabilities of the SS (only if present in the REG-REQ). The BS responds to the SS capabilities to indicate whether they may be used. If the BS does not recognize an SS capability, it shall return this as "off" in the REG-RSP.

Capabilities returned in the REG-RSP shall not be set to require greater capability of the SS than is indicated in the REG-REQ, as this is the handshake indicating that they have been successfully negotiated.

The following parameter may be included in the REG-RSP:

Vendor ID Encoding (of the BS; see 11.4.3)

The following parameter may be included in the REG-RSP if the REG-REQ contained the Vendor ID Encoding for the SS:

Vendor-specific extensions (see 11.4.11)

6.2.2.3.9 Privacy key management messages (PKM-REQ/PKM-RSP)

Privacy key management (PKM) employs two MAC message types, PKM Request (PKM-REQ), and PKM Response (PKM-RSP), as described in Table 22.

Table 22—PKM MAC messages

Type Value	Message name	Message description
9	PKM-REQ	Privacy Key Management Request [SS -> BS]
10	PKM-RSP	Privacy Key Management Response [BS -> SS]

While these two MAC management message types distinguish between PKM requests (SS–to–BS) and responses (BS–to–SS), more detailed information about message contents is encoded in the PKM messages themselves. This maintains a clean separation between privacy management functions and MAC bandwidth allocation, timing, and synchronization.

Exactly one PKM message is encapsulated in the Management Message Payload field of a MAC management message.

PKM protocol Messages transmitted from the SS to the BS shall use the form shown in Table 23. They are transmitted on the SSs Primary Management Connection.

Table 23—PKM request (PKM-REQ) message format

Syntax	Size	Notes
PKM-REQ_Message_Format() {		
Management Message Type = 9	8 bits	
Code	8 bits	
PKM Identifier	8 bits	
TLV Encoded Attributes	Variable	TLV specific
}		

PKM protocol Messages transmitted from the BS to the SS shall use the form shown in Table 24. They are transmitted on the SSs Primary Management Connection.

Table 24—PKM response (PKM-RSP) message format

Syntax	Size	Notes
PKM-RSP_Message_Format() {		
Management Message Type = 10	8 bits	
Code	8 bits	
PKM Identifier	8 bits	
TLV Encoded Attributes	Variable	TLV specific
}		

The parameters shall be as follows:

Code

The Code is one octet and identifies the type of PKM packet. When a packet is received with an invalid Code, it shall be silently discarded. The code values are defined in Table 25.

PKM Identifier

The Identifier field is one octet. An SS uses the identifier to match a BS response to the SS's requests.

The SS shall increment (modulo 256) the Identifier field whenever it issues a new PKM Message. A "new" Message is an Authorization Request or Key Request that is not a retransmission being sent in response to a Timeout event. For retransmissions, the Identifier field shall remain unchanged.

The Identifier field in Authentication Information Messages, which are informative and do not effect any response messaging, shall be set to zero. The Identifier field in a BS's PKM-RSP Message shall match the Identifier field of the PKM-REQ Message the BS is responding to. The Identifier field in Traffic Encryption Key (TEK) Invalid Messages, which are not sent in response to PKM-REQs, shall be set to zero. The Identifier field in unsolicited Authorization Invalid Messages shall be set to zero.

On reception of a PKM-RSP Message, the SS associates the Message with a particular state machine (the Authorization state machine in the case of Authorization Replies, Authorization Rejects, and Authorization Invalids; a particular TEK state machine in the case of Key Replies, Key Rejects, and TEK Invalids.

An SS shall keep track of the Identifier of its latest, pending Authorization Request. The SS shall discard Authorization Reply and Authorization Reject messages with Identifier fields not matching that of the pending Authorization Request.

An SS shall keep track of the Identifiers of its latest, pending Key Request for each security association (SA). The SS shall discard Key Reply and Key Reject messages with Identifier fields not matching those of the pending Key Request messages.

Attributes

PKM Attributes carry the specific authentication, authorization, and key management data exchanged between client and server. Each PKM packet type has its own set of required and optional Attributes. Unless explicitly stated, there are no requirements on the ordering of attributes within a PKM message. The end of the list of Attributes is indicated by the LEN field of the MAC PDU header.

Table 25—PKM message codes

Code	PKM Message Type	MAC Management Message Name
0-2	*reserved*	—
3	SA Add	PKM-RSP
4	Auth Request	PKM-REQ
5	Auth Reply	PKM-RSP
6	Auth Reject	PKM-RSP
7	Key Request	PKM-REQ
8	Key Reply	PKM-RSP
9	Key Reject	PKM-RSP
10	Auth Invalid	PKM-RSP
11	TEK Invalid	PKM-RSP
12	Authent Info	PKM-REQ
13—255	*reserved*	—

Formats for each of the PKM messages are described in the following subclauses. The descriptions list the PKM attributes contained within each PKM message type. The Attributes themselves are described in 11.2. Unknown attributes shall be ignored on receipt and skipped over while scanning for recognized attributes.

The BS shall silently discard all requests that do not contain ALL required attributes. The SS shall silently discard all responses that do not contain ALL required attributes.

6.2.2.3.9.1 Security Association Add (SA Add) message

This message is sent by the BS to the SS to establish one or more additional SAs.

Code: 3

Attributes are shown in Table 26.

Table 26—SA Add attributes

Attribute	Contents
Key-Sequence-Number	Authorization key sequence number
(one or more) SA-Descriptor(s)	Each compound SA-Descriptor Attribute specifies a security association identifier (SAID) and additional properties of the SA.

6.2.2.3.9.2 Authorization Request (Auth Request) message

Code: 4

Attributes are shown in Table 27.

Table 27—Auth Request attributes

Attribute	Contents
SS-Certificate	Contains the SS's X.509 user certificate
Security-Capabilities	Describes requesting SS's security capabilities
SAID	SS's primary SAID equal to the Basic CID

The SS-certificate attribute contains an X.509 SS certificate (see 7.6) issued by the SS's manufacturer. The SS's X.509 certificate is a public-key certificate which binds the SS's identifying information to its RSA public key in a verifiable manner. The X.509 certificate is digitally signed by the SS's manufacturer, and that signature can be verified by a BS that knows the manufacturer's public key. The manufacturer's public key is placed in an X.509 certification authority (CA) certificate, which in turn is signed by a higher level certification authority.

The Security-Capabilities attribute is a compound attribute describing the requesting SS's security capabilities. This includes the data encryption and data authentication algorithms the SS supports.

An SAID attribute contains a Privacy SAID. In this case, the provided SAID is the SS's Basic CID, which is equal to the Basic CID assigned to the SS during initial ranging.

6.2.2.3.9.3 Authorization Reply (Auth Reply) message

Sent by the BS to a client SS in response to an Authorization Request, the Authorization Reply message contains an Authorization Key, the key's lifetime, the key's sequence number, and a list of SA-Descriptors identifying the Primary and Static SAs the requesting SS is authorized to access and their particular properties (e.g., type, cryptographic suite). The Authorization Key shall be encrypted with the SS's public key. The SA-Descriptor list shall include a descriptor for the Basic CID reported to the BS in the corresponding Auth Request. The SA-Descriptor list may include descriptors of Static SAIDs the SS is authorized to access.

Code: 5

Attributes are shown in Table 28.

Table 28—Auth Reply attributes

Attribute	Contents
AUTH-Key	Authorization (AUTH) Key, encrypted with the target client SS's public key
Key-Lifetime	Authorization Key's active lifetime
Key-Sequence-Number	Authorization key sequence number
(one or more) SA-Descriptor(s)	Each compound SA-Descriptor Attribute specifies an SAID and additional properties of the SA.

6.2.2.3.9.4 Authorization Reject (Auth Reject) message

The BS responds to an SS's authorization request with an Authorization Reject message if the BS rejects the SS's authorization request.

Code: 6

Attributes are shown in Table 29.

Table 29—Auth Reject attributes

Attribute	Contents
Error-Code	Error code identifying reason for rejection of authorization request.
Display-String (optional)	Display String providing reason for rejection of authorization request.

The Error-Code and Display-String attributes describe to the requesting SS the reason for the authorization failure.

6.2.2.3.9.5 Key Request message

Code: 7

Attributes are shown in Table 30.

Table 30—Key Request attributes

Attribute	Contents
Key-Sequence-Number	Authorization key sequence number
SAID	Security Association ID
HMAC-Digest	Keyed secure hash algorithm (SHA) message digest

The HMAC Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the keyed digest allows the BS to authenticate the Key Request message. The HMAC-Digest's authentication key is derived from the Authorization Key. See 7.5 for details.

6.2.2.3.9.6 Key Reply message

Code: 8

Attributes are shown in Table 31.

Table 31—Key Reply attributes

Attribute	Contents
Key-Sequence-Number	Authorization key sequence number
SAID	Security Association ID
TEK-Parameters	"Older" generation of key parameters relevant to SAID
TEK-Parameters	"Newer" generation of key parameters relevant to SAID
HMAC-Digest	Keyed SHA message digest

The TEK-Parameters Attribute is a compound attribute containing all of the keying material corresponding to a particular generation of an SAID's TEK. This would include the TEK, the TEK's remaining key lifetime, its key sequence number, and the cipher block chaining (CBC) initialization vector. The TEK is encrypted. See 11.2.8 for details.

At all times the BS maintains two sets of active generations of keying material per SAID. (A set of keying material includes a TEK and its corresponding CBC initialization vector.) One set corresponds to the "older" generation of keying material, the second set corresponds to the "newer" generation of keying material. The newer generation has a key sequence number one greater than (modulo 4) that of the older generation. 7.4.1 specifies BS requirements for maintaining and using an SAID's two active generations of keying material.

The BS distributes to a client SS both generations of active keying material. Thus, the Key Reply message contains two TEK-Parameters Attributes, each containing the keying material for one of the SAID's two active sets of keying material.

The HMAC Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the keyed digest allows the receiving client to authenticate the Key Reply message and ensure SS and BS have synchronized Authorization Keys. The HMAC-Digest's authentication key is derived from the Authorization Key. See 7.5 for details.

6.2.2.3.9.7 Key Reject message

Receipt of a Key Reject indicates the receiving client SS is no longer authorized for a particular SAID.

Code: 9

Attributes are shown in Table 32.

Table 32—Key Reject attributes

Attribute	Contents
Key-Sequence-Number	Authorization key sequence number
SAID	Security Association ID
Error-Code	Error code identifying reason for rejection of Key Request
Display-String (optional)	Display string containing reason for Key Reject
HMAC-Digest	Keyed SHA message digest

The HMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the keyed digest allows the receiving client to authenticate the Key Reject message and ensure SS and BS have synchronized Authorization Keys. The HMAC-Digest's authentication key is derived from the Authorization Key. See 7.5 for details.

6.2.2.3.9.8 Authorization Invalid message

The BS may send an Authorization Invalid message to a client SS as:

a) an unsolicited indication, or

b) a response to a message received from that SS.

In either case, the Authorization Invalid message instructs the receiving SS to reauthorize with its BS.

The BS sends an Authorization Invalid in response to a Key Request if (1) the BS does not recognize the SS as being authorized (i.e., no valid Authorization Key associated with the requesting SS) or (2) verification of the Key Request's keyed message digest (in HMAC-Digest Attribute) failed, indicating a loss of Authorization Key synchronization between SS and BS.

Code: 10

Attributes are shown in Table 33.

Table 33—Authorization Invalid attributes

Attribute	Contents
Error-Code	Error code identifying reason for Authorization Invalid
Display-String (optional)	Display String describing failure condition

6.2.2.3.9.9 TEK Invalid message

The BS sends a TEK Invalid message to a client SS if the BS determines that the SS encrypted an uplink PDU with an invalid TEK (i.e., an SAID's TEK key sequence number), contained within the received packet's MAC Header, is out of the BS's range of known, valid sequence numbers for that SAID.

Code: 11

Attributes are shown in Table 34.

Table 34—TEK Invalid attributes

Attribute	Contents
Key-Sequence-Number	Authorization key sequence number
SAID	Security Association ID
Error-Code	Error code identifying reason for TEK Invalid message
Display-String (optional)	Display string containing vendor-defined information
HMAC-Digest	Keyed SHA message digest

The HMAC-Digest attribute shall be the final attribute in the message's attribute list.

Inclusion of the keyed digest allows the receiving client to authenticate the TEK Invalid message and ensure SS and BS have synchronized Authorization Keys. The HMAC-Digest's authentication key is derived from the Authorization Key. See 7.5 for details.

6.2.2.3.9.10 Authentication Information (Authent Info) message

The Authent Info message contains a single CA-Certificate Attribute, containing an X.509 CA certificate for the manufacturer of the SS. The SS's X.509 user certificate shall have been issued by the certification authority identified by the X.509 CA certificate.

Authent Info messages are strictly informative; while the SS shall transmit Authent Info messages as indicated by the Authentication state model (7.2.4), the BS may ignore them.

Code: 12

Attributes are shown in Table 35.

Table 35—Authent Info attributes

Attribute	Contents
CA-Certificate	Certificate of manufacturer CA that issued SS certificate

The CA-certificate attribute contains an X.509 CA certificate for the CA that issued the SS's X.509 user certificate. The external certification authority issues these CA certificates to SS manufacturers.

6.2.2.3.10 Dynamic Service Addition—Request (DSA-REQ) message

A DSA-REQ is sent by an SS or BS to create a new service flow.

Table 36—DSA-REQ message format

Syntax	Size	Notes
DSA-REQ_Message_Format() {		
Management Message Type = 11	8 bits	
Transaction ID	16 bits	
TLV Encoded Information	Variable	TLV specific
}		

An SS or BS shall generate DSA-REQ Messages in the form shown in Table 36, including the following parameters:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID.

Transaction ID
Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples.

A DSA-REQ Message shall not contain parameters for more than one service flow in each direction, i.e., a DSA-REQ Message shall contain parameters for either a single uplink service flow, or for a single downlink service flow, or for one uplink and one downlink service flow.

The DSA-REQ Message shall contain the following:

Service Flow Parameters (see 11.4.8)
Specification of the service flow's traffic characteristics and scheduling requirements.

Convergence Sublayer Parameter Encodings (see 11.4.9)
Specification of the service flow's CS specific parameters

If Privacy is enabled, the DSA-REQ Message shall contain the following:

HMAC Tuple (see 11.4.10)
The HMAC Tuple Attribute contains a keyed Message digest (to authenticate the sender). The HMAC Tuple Attribute shall be the final Attribute in the Dynamic Service Message's Attribute list.

6.2.2.3.10.1 SS-Initiated dynamic service addition (DSA)

Values of the service flow Reference are local to the DSA Message; each service flow within the DSA-REQ shall be assigned a unique service flow Reference. This value need not be unique with respect to the other service flows known by the sender.

SS-initiated DSA-REQs may use the Service Class Name in place of some, or all, of the QoS Parameters.

6.2.2.3.10.2 BS-Initiated DSA

BS-initiated DSA-REQs shall also include a CID. CIDs are unique within the MAC domain.

BS-initiated DSA-REQs for named Service Classes shall include the QoS Parameter Set associated with that Service Class. BS-initiated DSA-REQs shall also include the SA-Descriptor for the service flow.

6.2.2.3.11 Dynamic Service Addition—Response (DSA-RSP) message

A DSA-RSP shall be generated in response to a received DSA-REQ. The format of a DSA-RSP shall be as shown in Table 37.

Table 37—DSA-RSP message format

Syntax	Size	Notes
DSA-RSP_Message_Format() {		
Management Message Type = 12	8 bits	
Transaction ID	16 bits	
Confirmation Code	8 bits	
TLV Encoded Information	Variable	TLV specific
}		

Parameters shall be as follows:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID.

Transaction ID
Transaction ID from corresponding DSA-REQ.

Confirmation Code (see 11.4.12)
The appropriate Confirmation Code for the entire corresponding DSA-REQ.

All other parameters are coded as TLV tuples.

If the transaction is successful, the DSA-RSP may contain the following:

Service Flow Parameters (see 11.4.8)
The complete specification of the service flow shall be included in the DSA-RSP only if it includes a newly assigned CID or an expanded Service Class Name.

CS Parameter Encodings (see 11.4.9)
Specification of the service flow's CS specific parameters.

If the transaction is unsuccessful, the DSA-RSP shall include:

Service Flow Error Set (see 11.4.8.4)
A Service Flow Error Set and identifying Service Flow Reference/Identifier shall be included for every failed service flow in the corresponding DSA-REQ Message. Every Service Flow Error Set

shall include every specific failed QoS Parameter of the corresponding service flow (see 11.4.8). This parameter shall be omitted if the entire DSA-REQ is successful.

If Privacy is enabled, the DSA-RSP Message shall contain:

HMAC Tuple (see 11.4.10)
The HMAC Tuple Attribute contains a keyed Message digest (to authenticate the sender). The HMAC Tuple Attribute shall be the final Attribute in the Dynamic Service Message's Attribute list.

6.2.2.3.11.1 SS-Initiated DSA

The BS's DSA-RSP for service flows that are successfully added shall contain a CID. The DSA-RSP for successfully Admitted or Active uplink QoS Parameter Sets shall also contain a CID.

The BS's DSA-RSP shall also include the SA-Descriptor for the service flow. If the corresponding DSA-REQ uses the Service Class Name (see 11.4.8.3) to request service addition, a DSA-RSP shall contain the QoS Parameter Set associated with the named Service Class. If the Service Class Name is used in conjunction with other QoS Parameters in the DSA-REQ, the BS shall accept or reject the DSA-REQ using the explicit QoS Parameters in the DSA-REQ. If these service flow encodings conflict with the Service Class attributes, the BS shall use the DSA-REQ values as overrides for those of the Service Class.

If the transaction is unsuccessful, the BS shall use the original Service Flow Reference to identify the failed parameters in the DSA-RSP.

6.2.2.3.11.2 BS-Initiated DSA

If the transaction is unsuccessful, the SS shall use the CID to identify the failed parameters in the DSA-RSP.

6.2.2.3.12 Dynamic Service Addition—Acknowledge (DSA-ACK) message

A DSA-ACK shall be generated in response to a received DSA-RSP. The format of a DSA-ACK shall be as shown in Table 38.

Table 38—DSA-ACK message format

Syntax	Size	Notes
DSA-ACK_Message_Format() {		
Management Message Type = 13	8 bits	
Transaction ID	16 bits	
Confirmation Code	8 bits	
TLV Encoded Information	Variable	TLV specific
}		

Parameters shall be as follows:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID.

Transaction ID
Transaction ID from corresponding DSA-RSP.

Confirmation Code (see 11.4.12)
The appropriate Confirmation Code for the entire corresponding DSA-RSP.

All other parameters are coded TLV tuples.

Service Flow Error Set (see 11.4.8.4)
The Service Flow Error Set of the DSA-ACK Message encodes specifics of any failed service flows in the DSA-RSP Message. A Service Flow Error Set and identifying Service Flow Reference shall be included for every failed QoS Parameter of every failed service flow in the corresponding DSA-REQ Message (see 11.4.8). This parameter shall be omitted if the entire DSA-REQ is successful.

If Privacy is enabled, the DSA-ACK Message shall contain:

HMAC Tuple (see 11.4.10)
The HMAC Tuple Attribute contains a keyed Message digest (to authenticate the sender). The HMAC Tuple Attribute shall be the final Attribute in the Dynamic Service Message's Attribute list.

6.2.2.3.13 DSC-REQ—Request (DSC-REQ) message

A DSC-REQ is sent by an SS or BS to dynamically change the parameters of an existing service flow.

An SS or BS shall generate DSC-REQ Messages in the form shown in Table 39, including the following parameters:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID.

Transaction ID
Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples.

Table 39—DSC-REQ message format

Syntax	Size	Notes
DSC-REQ_Message_Format() {		
Management Message Type = 14	8 bits	
Transaction ID	16 bits	
TLV Encoded Information	Variable	TLV specific
}		

A DSC-REQ Message shall not carry parameters for more than one service flow in each direction, i.e., a DSC-REQ Message shall contain parameters for either a single uplink service flow, or for a single downlink service flow, or for one uplink and one downlink service flow. A DSC-REQ shall contain the following:

Service Flow Parameters (see 11.4.8)
Specification of the service flow's new traffic characteristics and scheduling requirements. The Admitted and Active Quality of Service Parameter Sets currently in use by the service flow. If the DSC Message is successful and it contains service flow parameters, but does not contain replacement sets for both Admitted and Active Quality of Service Parameter Sets, the omitted set(s) shall be set to null. If included, the service flow Parameters shall contain a Service Flow Identifier.

If Privacy is enabled, a DSC-REQ shall also contain:

HMAC Tuple (see 11.4.10)
The HMAC Tuple Attribute contains a keyed Message digest (to authenticate the sender). The HMAC Tuple Attribute shall be the final Attribute in the Dynamic Service Message's Attribute list.

6.2.2.3.14 Dynamic Service Change—Response (DSC-RSP) message

A DSC-RSP shall be generated in response to a received DSC-REQ. The format of a DSC-RSP shall be as shown in Table 40.

Table 40—Dynamic Service Change Response (DSC-RSP) message format

Syntax	Size	Notes
DSC-RSP_Message_Format() {		
Management Message Type = 15	8 bits	
Transaction ID	16 bits	
Confirmation Code	8 bits	
TLV Encoded Information	Variable	TLV Specific
}		

Parameters shall be as follows:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID

Transaction ID
Transaction ID from corresponding DSC-REQ

Confirmation Code (see 11.4.12)
The appropriate Confirmation Code for the corresponding DSC-REQ.

All other parameters are coded as TLV tuples.

If the transaction is successful, the DSC-RSP may contain the following:

Service Flow Parameters (see 11.4.8)
The complete specification of the service flow shall be included in the DSC-RSP only if it includes a newly assigned CID or an expanded Service Class Name. If a Service Flow Parameter set contained an uplink Admitted QoS Parameter Set and this service flow does not have an associated CID, the DSC-RSP shall include a CID. If a Service Flow Parameter set contained a Service Class Name and an Admitted QoS Parameter Set, the DSC-RSP shall include the QoS Parameter Set corresponding to the named Service Class. If specific QoS Parameters were also included in the Classed service flow request, these QoS Parameters shall be included in the DSC-RSP instead of any QoS Parameters of the same type of the named Service Class.

CS Parameter Encodings (see 11.4.9)
Specification of the service flow's CS specific parameters.

If the transaction is unsuccessful, the DSC-RSP shall contain the following:

Service Flow Error Set (see 11.4.8.4)
A Service Flow Error Set and identifying CID shall be included for every failed service flow in the corresponding DSC-REQ Message. Every Service Flow Error Set shall include every specific failed QoS Parameter of the corresponding service flow (see 11.4.8). This parameter shall be omitted if the entire DSC-REQ is successful.

Regardless of success or failure, if Privacy is enabled for the SS, the DSC-RSP shall contain:

HMAC Tuple (see 11.4.10)
The HMAC Tuple Attribute contains a keyed Message digest (to authenticate the sender). The HMAC Tuple Attribute shall be the final Attribute in the Dynamic Service Message's Attribute list).

6.2.2.3.15 Dynamic Service Change—Acknowledge (DSC-ACK) message

A DSC-ACK shall be generated in response to a received DSC-RSP. The format of a DSC-ACK shall be as shown in Table 41.

Table 41—DSC-ACK message format

Syntax	Size	Notes
DSC-ACK_Message_Format() {		
Management Message Type = 16	8 bits	
Transaction ID	16 bits	
Confirmation Code	8 bits	
TLV Encoded Information	Variable	TLV specific
}		

Parameters shall be as follows:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID.

Transaction ID
Transaction ID from the corresponding DSC-REQ.

Confirmation Code (see 11.4.12)
The appropriate Confirmation Code for the entire corresponding DSC-RSP.

All other parameters are coded TLV tuples.

Service Flow Error Set (see 11.4.8.4)
The Service Flow Error Set of the DSC-ACK Message encodes specifics of any failed service flows in the DSC-RSP Message. A Service Flow Error Set and identifying Service Flow Identifier shall be included for every failed QoS Parameter of each failed service flow in the corresponding DSC-RSP Message (see 11.4.8). This parameter shall be omitted if the entire DSC-RSP is successful.

If Privacy is enabled, the DSC-ACK Message shall contain the following:

HMAC Tuple (see 11.4.10)
The HMAC Tuple Attribute contains a keyed Message digest (to authenticate the sender). The HMAC Tuple Attribute shall be the final Attribute in the Dynamic Service Message's Attribute list.

6.2.2.3.16 Dynamic Service Deletion—Request (DSD-REQ) message

A DSD-REQ is sent by an SS or BS to delete an existing service flow. The format of a DSD-REQ shall be as shown in Table 42.

Table 42—DSD-REQ message format

Syntax	Size	Notes
DSD-REQ_Message_Format() {		
Management Message Type = 17	8 bits	
Transaction ID	16 bits	
Service Flow ID	32 bits	
TLV Encoded Information	Variable	TLV specific
}		

Parameters shall be as follows:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID.

Service Flow Identifier
The SFID to be deleted.

Transaction ID
Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples.

If Privacy is enabled, the DSD-REQ shall include the following:

HMAC Tuple (see 11.4.10)
The HMAC Tuple Attribute contains a keyed Message digest (to authenticate the sender). The HMAC Tuple Attribute shall be the final Attribute in the Dynamic Service Message's Attribute list.

6.2.2.3.17 Dynamic Service Deletion—Response (DSD-RSP) message

A DSD-RSP shall be generated in response to a received DSD-REQ. The format of a DSD-RSP shall be as shown in Table 43.

Table 43—DSD-RSP message format

Syntax	Size	Notes
DSD-RSP_Message_Format() {		
Management Message Type = 18	8 bits	
Transaction ID	16 bits	
Confirmation Code	8 bits	
Service Flow ID	32 bits	
TLV Encoded Information	Variable	TLV specific
}		

Parameters shall be as follows:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID.

Service Flow Identifier
SFID from the DSD-REQ to which this acknowledgement refers.

Transaction ID
Transaction ID from the corresponding DSD-REQ.

Confirmation Code (see 11.4.12)
The appropriate Confirmation Code for the corresponding DSD-REQ.

All other parameters are coded as TLV tuples.

If Privacy is enabled, the DSD-RSP shall include:

HMAC Tuple (see 11.4.10)
The HMAC Tuple Attribute contains a keyed Message digest (to authenticate the sender). The HMAC Tuple Attribute shall be the final Attribute in the Dynamic Service Message's Attribute list.

6.2.2.3.18 Multicast Polling Assignment Request (MCA-REQ) message

The MCA-REQ message is sent to an SS to assign it to or remove it from a multicast polling group. The format of the message is shown in Table 44.

Table 44—MCA-REQ message format

Syntax	Size	Notes
MCA-REQ_Message_Format() {		
Management Message Type = 21	8 bits	
Transaction ID	16 bits	
TLV Encoded Information	Variable	TLV specific
}		

Parameters shall be as follows:

CID *(in the Generic MAC Header)*
SS's Primary Management CID.

Transaction ID
Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples.

Multicast CID (See 11.1.5)
Assignment (See 11.1.5)

6.2.2.3.19 Multicast Polling Assignment Response (MCA-RSP) message

The MCA-RSP is sent by the SS in response to a MCA-REQ. The message format shall be as shown in Table 45.

Table 45—MCA-RSP message format

Syntax	Size	Notes
MCA-RSP_Message_Format() {		
Management Message Type = 22	8 bits	
Transaction ID	16 bits	
Confirmation Code	8 bits	
}		

Parameters shall be as follows:

CID ***(in the Generic MAC Header)***
SS's Primary Management CID.

Transaction ID
Unique identifier for this transaction assigned by the sender.

Confirmation Code

6.2.2.3.20 Downlink Burst Profile Change Request (DBPC-REQ) message

The DBPC-REQ Message is sent by the SS to the BS on the SS's Basic CID to request a change of the downlink burst profile used by the BS to transport data to the SS. Note that a change of downlink burst profile may also be requested by means of a RNG-REQ message as defined in 6.2.2.3.5.

The DBPC-REQ Message shall be sent at the current operational Data Grant Burst Type for the SS. If the SS has been inactive on its uplink for some period of time and detects fading on the downlink, the SS uses this message to request transition to a more robust Data Grant Burst Type. The message format shall be as shown in Table 46.

Table 46—DBPC-REQ message format

Syntax	Size	Notes
DBPC-REQ_Message_Format() {		
Management Message Type = 23	8 bits	
reserved	4 bits	bits reserved for future use
DIUC	4 bits	
}		

Parameters shall be as follows:

DIUC
DIUCs as defined in Table 91.

6.2.2.3.21 Downlink Burst Profile Change Response (DBPC-RSP) message

The DBPC-RSP Message shall be transmitted by the BS on the SS's Basic CID in response to a DBPC-REQ message from the SS. If the DIUC parameter is the same as requested in the DBPC-REQ message, then the request was accepted. Otherwise, if the request is rejected, the DIUC parameter shall be the previous DIUC at which the SS was receiving downlink data. The message format shall be as shown in Table 47.

Table 47—DBPC-RSP message format

Syntax	Size	Notes
DBPC-RSP_Message_Format() {		
Management Message Type = 24	8 bits	
reserved	4 bits	bits reserved for future use
DIUC	4 bits	
}		

Parameters shall be as follows:

DIUC
DIUCs as defined in Table 91.

6.2.2.3.22 Reset Command (RES-CMD) message

The RES-CMD Message shall be transmitted by the BS on an SS's Basic CID to force the SS to reset itself, reinitialize its MAC, and repeat initial system access. This message may be used if an SS is unresponsive to the BS or if the BS detects continued abnormalities in the uplink transmission from the SS.

The MAC Management Message type for this message is given in Table 3. The RES-CMD Message format is shown in Table 48.

Table 48—RES-CMD message format

Syntax	Size	Notes
RES-CMD_Message_Format() {		
Management Message Type = 25	8 bits	
TLV encoded information	variable	
}		

The RES-CMD shall include the following parameters encoded as TLV tuples:

HMAC Tuple (see 11.4.12)
The HMAC Tuple shall be the last attribute in the message.

6.2.2.3.23 SS Basic Capability Request (SBC-REQ) message

The SS SBC-REQ shall be transmitted by the SS during initialization. An SS shall generate SBC-REQ messages in the form shown in Table 49.

Table 49—SS SBC-REQ message format

Syntax	Size	Notes
SBC-REQ_Message_Format() {		
Management Message Type = 26	8 bits	
TLV Encoded Information	Variable	TLV specific
}		

An SS shall generate SS SBC-REQs including the following parameter:

Basic CID (in the MAC Header)
The CID in the MAC Header is the Basic CID for this SS, as assigned in the RNG-RSP message.

All other parameters are coded as TLV tuples.

Basic Capability Requests contain those SS Capabilities Encodings (11.4.1) that are necessary for effective communication with the SS during the remainder of the initialization protocols. Only the following parameters shall be included in the Basic Capabilities Request:

Physical Parameters Supported (see 11.4.1.2)
Bandwidth Allocation Support (see 11.4.1.6)

6.2.2.3.24 SS Basic Capability Response (SBC-RSP) message

The SS SBC-RSP shall be transmitted by the BS in response to a received SBC-REQ.

To provide flexibility, the message parameters following the Response field shall be encoded in a TLV format.

Table 50—SS SBC-RSP message format

Syntax	Size	Notes
SBC-RSP_Message_Format() {		
Management Message Type = 27	8 bits	
Confirmation Code	8 bits	
TLV Encoded Attributes	Variable	TLV specific
}		

A BS shall generate SS SBC-RSPs in the form shown in Table 50, including both of the following parameters:

CID (in the MAC Header)
CID from corresponding SBC-REQ to which this response refers (this acts as a transaction identifier).

The following parameters shall be included in the SBC-RSP if found in the SS SBC-REQ:

Physical Parameters Supported (see 11.4.1.2)

Bandwidth Allocation Support (see 11.4.1.6)
The BS response to the subset of SS capabilities present in the SBC-REQ message. The BS responds to the SS capabilities to indicate whether they may be used. If the BS does not recognize an SS capability, it shall return this as “off” in the SBC-RSP.

Only capabilities set to “on” in the SBC-REQ may be set “on” in the REG-RSP, as this is the handshake indicating that they have been successfully negotiated.

6.2.2.3.25 Clock Comparison (CLK-CMP) message

In network systems with service flows carrying information that requires the SSs to reconstruct their network clock signals (e.g., DS1 and DS3), CLK-CMP messages shall be periodically broadcast by the BS.

If provisioned to do so, the BS shall generate one CLK-CMP message at every periodic interval defined in Table 118 according to the format shown in Table 51.

Table 51—CLK-CMP message format

Syntax	Size	Notes
CLK-CMP_Message_Format() {		
Management Message Type = 28	8 bits	
Clock Count *n*	8 bits	
for (*i* = 1; *i* <= *n*; *i*++) {		For each clock signal 1 through *n*
Clock ID[*i*]	8 bits	
Sequence Number[*i*]	8 bits	
Comparison Value[*i*]	8 bits	
}		
}		

CLK-CMP messages shall include the following parameters where Clock ID, Sequence Number, and Clock Comparison Value (CCV) shall be repeated for each clock signal:

Clock Count
This 8-bit value shall be the number of CCVs included in the CLK-CMP message.

Clock ID
This 8-bit value shall be the unique identifier for each clock signal from which the CCVs are generated by the BS.

Sequence Number
This 8-bit value shall be incremented by one (modulo the field size, 256) by the BS whenever a new CLK-CMP message is generated. This parameter is used to detect packet losses.

Clock Comparison Value
This 8-bit value shall be the difference (modulo the field size, 256) between the following two reference clock signals: (1) a 10 MHz reference clock locked to the symbol clock of the airlink (such as a GPS reference used to generate the symbol clock), and (2) an 8.192 MHz reference clock locked to the network clock.

6.2.2.3.26 De/Re-register Command (DREG-CMD) message

The DREG-CMD Message shall be transmitted by the BS on an SS's Basic CID to force the SS to change its access state. Upon receiving a DREG-CMD, the SS shall take the action indicated by the action code.

The MAC Management Message type for this message is given in Table 13. The format of the message is shown in Table 52.

Table 52—DREG-CMD message format

Syntax	Size	Notes
DREG-CMD_Message_Format() {		
Management Message Type = 29	8 bits	
Action Code	8 bits	
TLV encoded parameters	variable	
}		

The Action Code values and the corresponding actions are specified in Table 53.

Table 53—Action Codes and actions

Action Code	Action
0x00	SS shall leave the current channel and attempt to access another channel.
0x01	SS shall listen to the current channel but shall not transmit until an RES-CMD message is received.
0x02	SS shall listen to the current channel but only transmit on the Basic, Primary Management, and Secondary Management Connections.
0x03	SS shall return to normal operation and may transmit on any of its active connections.
0x04-0xFF	*reserved*

The DREG-CMD shall include the following parameters encoded as TLV tuples:

HMAC Tuple (see 11.4.10)
The HMAC Tuple shall be the last attribute in the message.

6.2.2.3.27 DSx Received (DSX-RVD) message

The Dynamic Service Message Received message shall be generated by the BS in response to an SS-initiated DSx-REQ to inform the SS that the BS has received the DSx-REQ message in a more timely manner than provided by the DSx-RSP message, which shall be transmitted only after the DSx-REQ is authenticated. The format of the DSX-RVD shall be as shown in Table 54.

Table 54—DSX-RVD message format

Syntax	Size	Notes
DSX-RVD_Message_Format() {		
Management Message Type = 30	8 bits	
Transaction ID	16 bits	
Confirmation Code	8 bits	
}		

Parameters shall be as follows:

CID (in the Generic MAC Header)
SS's Primary Management CID.

Transaction ID
Transaction ID from corresponding DSA-REQ.

Confirmation Code (see 11.4.12)
The appropriate Confirmation Code indicating the integrity of the corresponding DSx-REQ.

6.2.2.3.28 Config File TFTP Complete (TFTP-CPLT) message

The Config File TFTP-CPLT Message shall be generated by the SS when it has successfully retrieved its configuration file from the provisioning server. The format of the TFTP-CPLT shall be as shown in Table 55.

Table 55—TFTP-CPLT message format

Syntax	Size	Notes
TFTP-CPLT_Message_Format() {		
Management Message Type = 31	8 bits	
TLV encoded information	variable	
}		

Parameters shall be as follows:

CID (in the Generic MAC Header)
SS's Primary Management CID.

The TFTP-CPLT shall include the following parameters encoded as TLV tuples:

HMAC Tuple (see 11.4.10)
The HMAC Tuple shall be the last attribute in the message.

6.2.2.3.29 Config File TFTP Complete Response (TFTP-RSP) message

The Config File TFTP-RSP Message shall be generated by the BS in response to a TFTP-CPLT message from the SS. The format of the TFTP-RSP shall be as shown in Table 56.

Table 56—Config File TFTP-RSP message format

Syntax	Size	Notes
TFTP-RSP_Message_Format() {		
Management Message Type = 32	8 bits	
}		

Parameters shall be as follows:

CID (in the Generic MAC Header)
SS's Primary Management CID.

6.2.3 Construction and transmission of MAC PDUs

The construction of a MAC PDU is illustrated in Figure 25.

6.2.3.1 Conventions

Messages are always transmitted in the order: Most Significant Byte first, with the Most Significant Bit first in each byte.

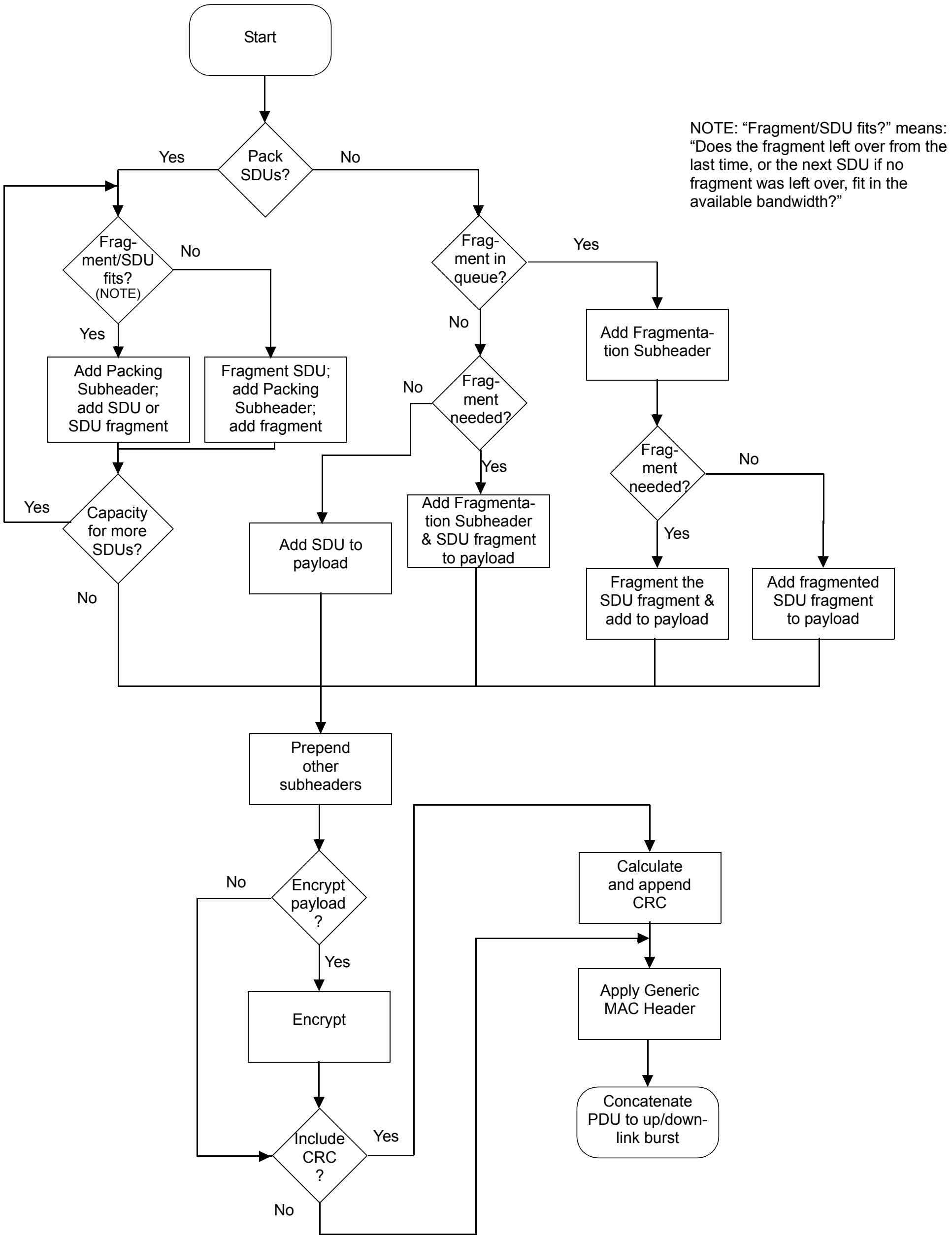

Start
Pack SDUs?
Yes
No
NOTE: "Fragment/SDU fits?" means: "Does the fragment left over from the last time, or the next SDU if no fragment was left over, fit in the available bandwidth?"
Frag-ment/SDU fits? (NOTE)
No
Yes
Frag-ment in queue?
Yes
No
Add Packing Subheader; add SDU or SDU fragment
Fragment SDU; add Packing Subheader; add fragment
Add Fragmenta-tion Subheader
Frag-ment needed?
No
Yes
Add SDU to payload
Add Fragmenta-tion Subheader & SDU fragment to payload
Frag-ment needed?
No
Yes
Capacity for more SDUs?
Yes
No
Fragment the SDU fragment & add to payload
Add fragmented SDU fragment to payload
Prepend other subheaders
Encrypt payload ?
No
Yes
Encrypt
Include CRC ?
Yes
No
Calculate and append CRC
Apply Generic MAC Header
Concatenate PDU to up/down-link burst

Figure 25—Construction of a MAC PDU

6.2.3.2 Concatenation

Multiple MAC PDUs may be concatenated into a single transmission in either the uplink or downlink directions. Figure 26 illustrates this concept for an uplink burst transmission. Since each MAC PDU is identified by a unique CID, the receiving MAC entity is able to present the MAC SDU (after reassembling the MAC SDU from one or more received MAC PDUs) to the correct instance of the MAC SAP. MAC Management messages, user data, and bandwidth request MAC PDUs may be concatenated into the same transmission.

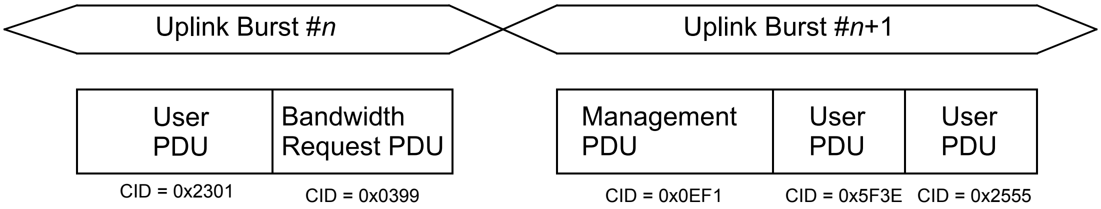


Figure 26—MAC PDU concatenation showing example CIDs

6.2.3.3 Fragmentation

Fragmentation is the process by which a MAC SDU is divided into one or more MAC PDUs. This process is undertaken to allow efficient use of available bandwidth relative to the QoS requirements of a connection's service flow.

Fragmentation may be initiated by a BS for a downlink connection. Fragmentation may be initiated by an SS for an uplink connection. A connection may be in only one fragmentation state at any given time.

The authority to fragment traffic on a connection is defined when the connection is created by the MAC SAP.

The fragments are set in accordance with Table 57.

Table 57—Fragmentation rules

Fragment	FC	FSN
First Fragment	10	Incremented modulo 8
Continuing Fragment	11	Incremented modulo 8
Last Fragment	01	Incremented modulo 8
Unfragmented	00	Incremented modulo 8

The sequence number allows the SS to recreate the original payload and to detect the loss of any intermediate packets. Upon loss, the SS shall discard all MAC PDUs on the connection until a new first fragment is detected or a nonfragmented MAC PDU is detected.

6.2.3.4 Packing

If packing is turned on for a connection, the MAC may pack multiple MAC SDUs into a single MAC PDU. Packing makes use of the connection attribute indicating whether the connection carries fixed-length or variable-length packets. The transmitting side has full discretion whether or not to pack a group of MAC SDUs in a single MAC PDU.

6.2.3.4.1 Packing fixed-length MAC SDUs

For connections that are indicated, by the fixed-length versus variable-length SDU indicator (11.4.8.15), to carry fixed-length MAC SDUs, the packing procedure described in this subclause may be used. In this case, the Request/Transmission Policy (11.4.8.12) shall be set to allow packing and prohibit fragmentation, and the SDU size (11.4.8.16) shall be included in DSA-REQ message when establishing the connection. The length field of the MAC header implicitly indicates the number of MAC SDUs packed into a single MAC

PDU. If the MAC SDU size is *n* bytes, the receiving side can unpack simply by knowing that the length field in the MAC header will be *n*k+j*, where *k* is the number of MAC SDUs packed into the MAC PDU and *j* is the size of the MAC header and any prepended MAC subheaders. A MAC PDU containing a packed sequence of fixed-length MAC SDUs would be constructed as in Figure 27. Note that there is no added overhead due to packing in the fixed-length MAC SDU case, and a single MAC SDU is simply a packed sequence of length 1.

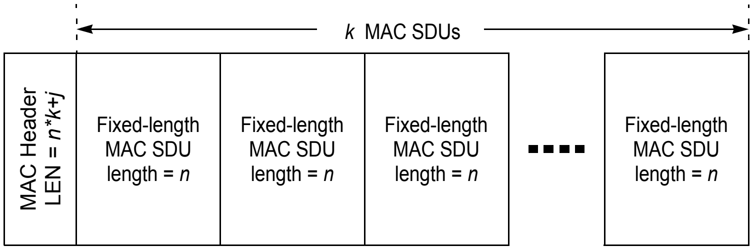
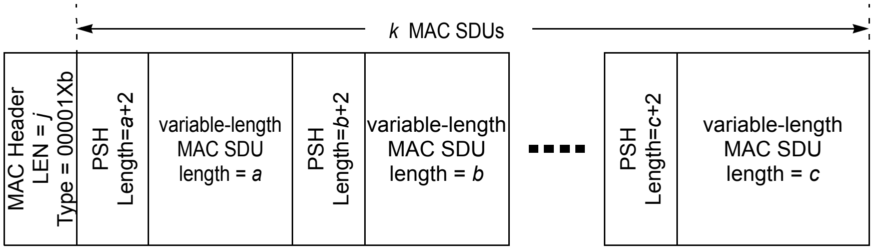

k MAC SDUs
MAC Header
LEN = *n*k+j*
Fixed-length MAC SDU length = *n*
Fixed-length MAC SDU length = *n*
Fixed-length MAC SDU length = *n*
Fixed-length MAC SDU length = *n*

Figure 27—Packing fixed-length MAC SDUs into a single MAC PDU

6.2.3.4.2 Packing variable-length MAC SDUs

When packing variable-length MAC SDU connections, such as Ethernet, the *n*k+j* relationship between the MAC header's length field and the higher-layer MAC SDUs no longer holds. This necessitates indication of where one MAC SDU ends and another begins. In the variable-length MAC SDU case, the MAC attaches a Packing subheader (PSH) to each MAC SDU. This subheader is described in 6.2.2.2.3.

A MAC PDU containing a packed sequence of variable-length MAC SDUs is constructed as shown in Figure 28. If more than one MAC SDU is packed into the MAC PDU, the type field in the MAC header indicates the presence of Packing subheaders. Note that unfragmented MAC SDUs and MAC SDU fragments may both be present in the same MAC PDU, as described in 6.2.3.4.2.1.


k MAC SDUs
MAC Header
LEN = *j*
Type = 00001Xb
PSH
Length=*a*+2
variable-length MAC SDU length = *a*
PSH
Length=*b*+2
variable-length MAC SDU length = *b*
PSH
Length=*c*+2
variable-length MAC SDU length = *c*

Note: If Type=0x03, a GM subheader (not shown in figure) is also present immediately following the Generic MAC header.

Figure 28—Packing variable-length MAC SDUs into a single MAC PDU

If only one MAC SDU (or fragment thereof) is present in a MAC PDU, the type field in MAC header indicates the absence of Packing subheaders. This is shown in Figure 29.

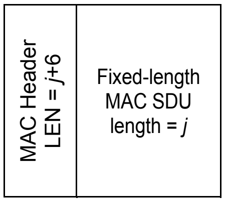
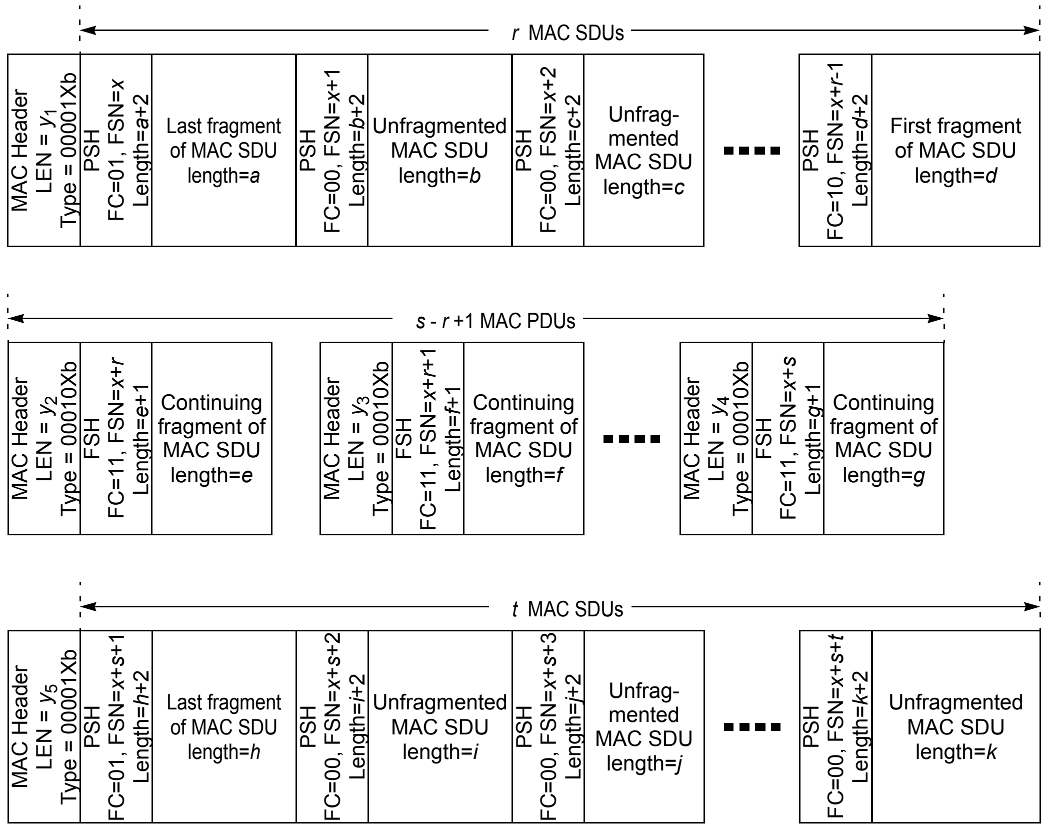


Figure 29—Packing a single variable-length MAC SDU into a single MAC PDU

6.2.3.4.2.1 Interaction with fragmentation

Simultaneous fragmentation and packing allows efficient use of the airlink, but requires guidelines to be followed so it is clear which MAC SDU is currently in a state of fragmentation. To accomplish this, when a Packing subheader is present, the fragmentation information for individual MAC SDUs or MAC SDU fragments is contained in the corresponding Packing subheader. If no Packing subheader is present, the fragmentation information for individual MAC SDU fragments is contained in the corresponding Fragmentation subheader. This is shown in Figure 30.


r MAC SDUs
MAC Header LEN = y1 Type = 00001Xb
PSH FC=01, FSN=x Length=a+2
Last fragment of MAC SDU length=a
PSH FC=00, FSN=x+1 Length=b+2
Unfragmented MAC SDU length=b
PSH FC=00, FSN=x+2 Length=c+2
Unfrag- mented MAC SDU length=c
▪▪▪▪
PSH FC=10, FSN=x+r-1 Length=d+2
First fragment of MAC SDU length=d
s - r +1 MAC PDUs
MAC Header LEN = y2 Type = 00010Xb
FSH FC=11, FSN=x+r Length=e+1
Continuing fragment of MAC SDU length=e
MAC Header LEN = y3 Type = 00010Xb
FSH FC=11, FSN=x+r+1 Length=f+1
Continuing fragment of MAC SDU length=f
▪▪▪▪
MAC Header LEN = y4 Type = 00010Xb
FSH FC=11, FSN=x+s Length=g+1
Continuing fragment of MAC SDU length=g
t MAC SDUs
MAC Header LEN = y5 Type = 00001Xb
PSH FC=01, FSN=x+s+1 Length=h+2
Last fragment of MAC SDU length=h
PSH FC=00, FSN=x+s+2 Length=i+2
Unfragmented MAC SDU length=i
PSH FC=00, FSN=x+s+3 Length=j+2
Unfrag- mented MAC SDU length=j
▪▪▪▪
PSH FC=00, FSN=x+s+t Length=k+2
Unfragmented MAC SDU length=k

Note: If Type=0x03, a GM subheader (not shown in figure) is also present immediately following the Generic MAC header.

Figure 30—Packing with fragmentation

Note that while it is legal to have continuation fragments packed with other fragments, the circumstances for creating continuation fragments would preclude this from happening.

6.2.3.5 CRC calculation

A service flow may require that it be carried on a connection in MAC PDUs with CRCs added (11.4.8.12). In this case, a CRC, as defined in IEEE 802.3, shall be included in each MAC PDU with HT=0; i.e., request MAC PDUs are unprotected. The CRC shall cover the Generic MAC Header and the Payload of the MAC PDU. The CRC shall be calculated after encryption; i.e. the CRC protects the Generic Header and the ciphered Payload.

6.2.3.6 Encryption of MAC PDUs

When transmitting a MAC PDU on a connection that is mapped to an SA, the sender shall perform encryption and data authentication of the MAC PDU payload as specified by that SA. When receiving a MAC PDU on a connection mapped to an SA, the receiver shall perform decryption and data authentication of the MAC PDU payload, as specified by that SA.

NOTE—Data authentication is not currently defined.

The Generic MAC Header shall not be encrypted. The Header contains all the Encryption information (Encryption Control Field, Encryption Key Sequence Field, and CID) needed to decrypt a Payload at the receiving station. This is illustrated in Figure 31.

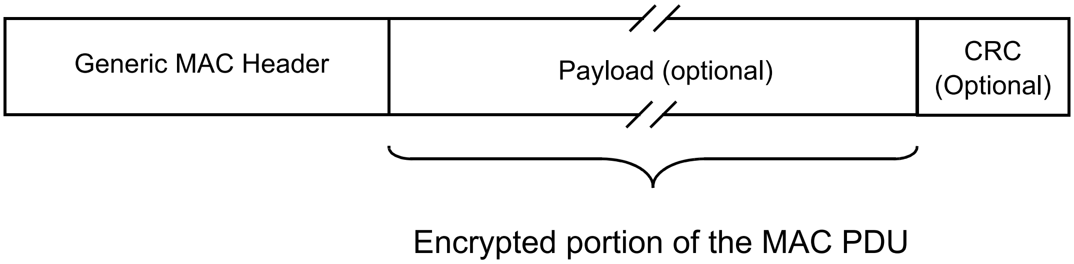


Figure 31—MAC PDU encryption

Two bits of a MAC Header contain a key sequence number. Note that the keying material associated with an SA has a limited lifetime, and the BS periodically refreshes an SA’s keying material. The BS manages a 2-bit key sequence number independently for each SA and distributes this key sequence number along with the SA’s keying material to the client SS. The BS increments the key sequence number with each new generation of keying material. The MAC Header includes this sequence number to identify the specific generation of that SA keying material being used to encrypt the attached payload. Being a 2-bit quantity, the sequence number wraps around to 0 when it reaches 3.

Comparing a received MAC PDU’s key sequence number with what it believes to be the “current” key sequence number, an SS or BS can easily recognize a loss of key synchronization with its peer. An SS shall maintain the two most recent generations of keying material for each SA. Keeping on hand the two most recent key generations is necessary for maintaining uninterrupted service during an SA’s key transition.

Encryption of the payload is indicated by the Encryption Control (EC) bit field. A value of 1 indicates the payload is encrypted and the EKS field contains meaningful data. A value of 0 indicates the payload is not encrypted. Any unencrypted MAC PDU received on a connection mapped to an SA requiring encryption shall be discarded.

6.2.4 ARQ mechanism

Automatic repeat request (ARQ) shall not be used with the PHY specification defined in 8.2.

6.2.5 Uplink scheduling service

Scheduling services are designed to improve the efficiency of the poll/grant process. By specifying a scheduling service and its associated QoS parameters, the BS can anticipate the throughput and latency needs of the uplink traffic and provide polls and/or grants at the appropriate times.

The basic services, as summarized in Table 58, are Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Non-Real-Time Polling Service (nrtPS) and Best Effort (BE) service. Each service is tailored to a specific type of data flow. The following subclauses define the basic uplink service flow scheduling services and list the QoS parameters associated with each service. A detailed description of each QoS parameter is provided in 11.4.8.

Table 58—Scheduling services and usage rules

Scheduling type	PiggyBack Request	Bandwidth stealing	Polling
UGS	Not Allowed	Not allowed	PM bit is used to request a unicast poll for bandwidth needs of non-UGS connections.
rtPS	Allowed	Allowed for GPSS	Scheduling only allows unicast polling.
nrtPS	Allowed	Allowed for GPSS	Scheduling may restrict a service flow to unicast polling via the transmission/request policy; otherwise all forms of polling are allowed.
BE	Allowed	Allowed for GPSS	All forms of polling allowed.

6.2.5.1 Unsolicited Grant Service

The Unsolicited Grant Service (UGS) is designed to support real-time service flows that generate fixed size data packets on a periodic basis, such as T1/E1 and Voice over IP without silence suppression. The service offers fixed size grants on a real-time periodic basis, which eliminate the overhead and latency of SS requests and assure that grants are available to meet the flow's real-time needs. The BS shall provide fixed size Data Grant Burst Types at periodic intervals to the service flow. In order for this service to work correctly, the Request/Transmission Policy (see 11.4.8.12) setting shall be such that the SS is prohibited from using any contention request opportunities, and the BS shall not provide any unicast request opportunities for that connection. This results in the SS only using unsolicited Data Grant Burst Types for uplink transmission on that connection. All other bits of the Request/Transmission Policy are irrelevant to the fundamental operation of this scheduling service and should be set according to network policy. The UGS shall be specified using the following parameters: the Unsolicited Grant Size, the Nominal Grant interval, the Tolerated Grant Jitter, and the Request/Transmission Policy.

The Grant Management subheader (6.2.2.2.2) is used to pass status information from the SS to the BS regarding the state of the UGS service flow. The most significant bit of the Grant Management field is the Slip Indicator (SI) bit. The SS shall set this flag once it detects that this service flow has exceeded its transmit queue depth. Once the SS detects that the service flow's transmit queue is back within limits, it shall clear the SI flag. The flag allows the BS to provide for long term compensation for conditions, such as lost maps or clock rate mismatches, by issuing additional grants. The poll-me bit (6.2.6.4.3) may be used to request to be polled for a different, non-UGS connection.

The BS shall not allocate more bandwidth than the Maximum Sustained Traffic Rate parameter of the Active QoS Parameter Set, excluding the case when the SI bit of the Grant Management field is set. In this case, the BS may grant up to 1% additional bandwidth for clock rate mismatch compensation.

6.2.5.2 Real-Time Polling Service

The Real-Time Polling Service (rtPS) is designed to support real-time service flows that generate variable size data packets on a periodic basis, such as MPEG video. The service offers real-time, periodic, unicast request opportunities, which meet the flow's real-time needs and allow the SS to specify the size of the desired grant. This service requires more request overhead than UGS, but supports variable grant sizes for optimum data transport efficiency.

The BS shall provide periodic unicast request opportunities. In order for this service to work correctly, the Request/Transmission Policy setting (see 11.4.8.12) shall be such that the SS is prohibited from using any contention request opportunities for that connection. The BS may issue unicast request opportunities as prescribed by this service even if a grant is pending. This results in the SS using only unicast request opportunities in order to obtain uplink transmission opportunities (the SS could still use unsolicited Data Grant Burst Types for uplink transmission as well). All other bits of the Request/Transmission Policy are irrelevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service information elements are the Nominal Polling Interval, the Tolerated Poll Jitter, and the Request/Transmission Policy.

6.2.5.3 Non-Real-Time Polling Service

The Non-Real-Time Polling Service (nrtPS) is designed to support non real-time service flows that require variable size Data Grant Burst Types on a regular basis, such as high bandwidth FTP. The service offers unicast polls on a regular basis, which assures that the flow receives request opportunities even during network congestion. The BS typically polls nrtPS CIDs on an interval (periodic or non-periodic) on the order of one second or less.

The BS shall provide timely unicast request opportunities. In order for this service to work correctly, the Request/Transmission Policy setting (see 11.4.8.12) should be such that the SS is allowed to use contention request opportunities. This results in the SS using contention request opportunities as well as unicast request opportunities and unsolicited Data Grant Burst Types. All other bits of the Request/Transmission Policy are irrelevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service elements are Nominal Polling Interval, Minimum Reserved Traffic Rate, Maximum Sustained Traffic Rate, Request/Transmission Policy, and Traffic Priority.

6.2.5.4 Best Effort service

The intent of the Best Effort (BE) service is to provide efficient service to best effort traffic. In order for this service to work correctly, the Request/Transmission Policy setting should be such that the SS is allowed to use contention request opportunities. This results in the SS using contention request opportunities as well as unicast request opportunities and unsolicited Data Grant Burst Types. All other bits of the Request/Transmission Policy are irrelevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service elements are the Minimum Reserved Traffic Rate, the Maximum Sustained Traffic Rate, and the Traffic Priority.

6.2.6 Bandwidth allocation and request mechanisms

Note that at registration every SS is assigned three dedicated CIDs for the purpose of sending and receiving control messages. Three connections are used to allow differentiated levels of QoS to be applied to the different connections carrying MAC management traffic. Increasing (or decreasing) bandwidth requirements is necessary for all services except incompressible constant bit rate UGS connections. The needs of incompressible UGS connections do not change between connection establishment and termination. The requirements of compressible UGS connections, such as channelized T1, may increase or decrease depending on traffic. Demand Assigned Multiple Access (DAMA) services are given resources on a demand assignment basis, as the need arises.

When an SS needs to ask for bandwidth on a connection with BE scheduling service, it sends a message to the BS containing the immediate requirements of the DAMA connection. QoS for the connection was established at connection establishment and is looked up by the BS.

There are numerous methods by which the SS can get the bandwidth request message to the BS.

6.2.6.1 Requests

Requests refer to the mechanism that SSs use to indicate to the BS that they need uplink bandwidth allocation. A Request may come as a stand-alone Bandwidth Request Header or it may come as a PiggyBack Request (see 6.2.2).

Because the uplink burst profile can change dynamically, all requests for bandwidth shall be made in terms of the number of bytes needed to carry the MAC header and payload, but not the PHY overhead. The Bandwidth Request Message may be transmitted during any of the following intervals:

Request IE
Any Data Grant Burst Type IE

Bandwidth Requests may be incremental or aggregate. When the BS receives an incremental Bandwidth Request, it shall add the quantity of bandwidth requested to its current perception of the bandwidth needs of the connection. When the BS receives an aggregate Bandwidth Request, it shall replace its perception of the bandwidth needs of the connection with the quantity of bandwidth requested. The Type field in the Bandwidth Request Header indicates whether the request is incremental or aggregate. Since Piggybacked Bandwidth Requests do not have a type field, Piggybacked Bandwidth Requests shall always be incremental. The self-correcting nature of the request/grant protocol requires that SSs shall periodically use aggregate Bandwidth Requests. The period may be a function of the QoS of a service and of the link quality. Due to the possibility of collisions, Bandwidth Requests transmitted in broadcast or multicast Request IEs should be aggregate requests.

Regarding the grant of the bandwidth requested, there are two modes of operation for SSs: Grant per Connection mode (GPC) and Grant per Subscriber Station mode (GPSS). In the first case, the BS grants bandwidth explicitly to each connection, whereas in the second case the bandwidth is granted to all the connections belonging to the SS. The latter case (GPSS) allows smaller uplink (UL) maps and allows more intelligent SSs to make last moment decisions and perhaps utilize the bandwidth differently than it was originally granted by the BS. This may be useful for real-time applications that require a faster response from the system.

Systems using the 10–66 GHz PHY specification in 8.2 shall use GPSS mode.

6.2.6.2 Grants per connection (GPC) mode

For an SS in GPC mode, the bandwidth requests are addressed explicitly to individual CIDs. Since it is non-deterministic which request is being honored, when the SS receives a shorter transmission opportunity than expected (i.e., scheduler decision, request message lost, etc.), no explicit reason is given. In all cases, based on the latest information received from the BS and the status of the request, the SS may decide to perform backoff and request again or to discard the MAC SDU.

A GPC SS may use Request IEs that are broadcast, directed at a multicast polling group it is a member of, or directed at a unicast CID that represents a service flow belonging to that SS. The burst profile associated with the Request IE shall be used, even if the BS is capable of receiving the SS with a more efficient burst profile. To take advantage of a more efficient burst profile, the SS should transmit in an interval defined by a Data Grant IE directed at a unicast CID that represents a service flow belonging to that SS. Because of this, unicast polling of a GPC SS would normally be done by allocating a Data Grant IE directed at a unicast CID that represents a service flow belonging to that SS. Also note that, in a Data Grant IE directed at a unicast CID that represents a service flow belonging to a GPC SS, the SS shall make bandwidth requests only for the indicated connection.

The procedure followed by SSs operating in GPC mode is shown in Figure 32.

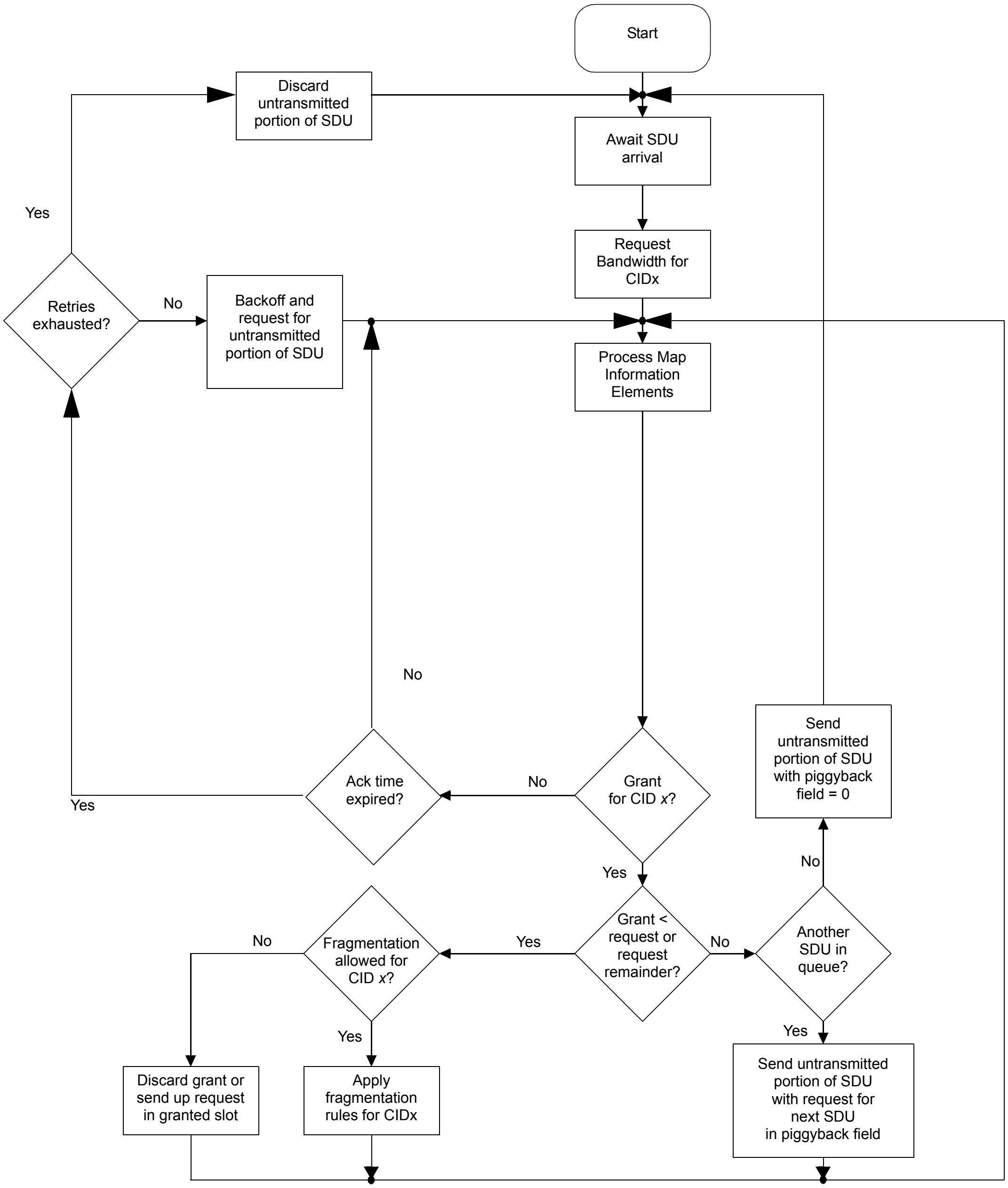

Start
Discard untransmitted portion of SDU
Await SDU arrival
Request Bandwidth for CIDx
Yes
Retries exhausted?
No
Backoff and request for untransmitted portion of SDU
Process Map Information Elements
No
Send untransmitted portion of SDU with piggyback field = 0
Yes
Ack time expired?
No
Grant for CID x?
Yes
No
No
Fragmentation allowed for CID x?
Yes
Grant < request or request remainder?
No
Another SDU in queue?
Yes
Yes
Discard grant or send up request in granted slot
Apply fragmentation rules for CIDx
Send untransmitted portion of SDU with request for next SDU in piggyback field

Figure 32—SS GPC mode flow chart

6.2.6.3 Grants per subscriber station (GPSS) mode

For an SS operating in GPSS mode, the bandwidth requests are addressed to the individual connections while the bandwidth grant is addressed to the SS's Basic CIDs and not explicitly to individual CIDs. Since it is nondeterministic which request is being honored, when the SS receives a shorter transmission opportunity than expected (i.e., scheduler decision, request message lost, etc.), no explicit reason is given. In all cases, based on the latest information received from the BS and the status of the request, the SS may decide to perform backoff and request again or to discard the SDU.

A GPSS SS may use Request IEs that are broadcast, directed at a multicast polling group it is a member of, or directed at its Basic CID. In all cases, the Request IE burst profile is used, even if the BS is capable of receiving the SS with a more efficient burst profile. To take advantage of a more efficient burst profile, the SS should transmit in an interval defined by a Data Grant IE directed at its Basic CID. Because of this, unicast polling of a GPSS SS would normally be done by allocating a Data Grant IE directed at its Basic CID. Also note that, in a Data Grant IE directed at its Basic CID, the SS may make bandwidth requests for any of its connections.

The procedure followed by SSs operating in GPSS mode is shown in Figure 33.

6.2.6.4 Polling

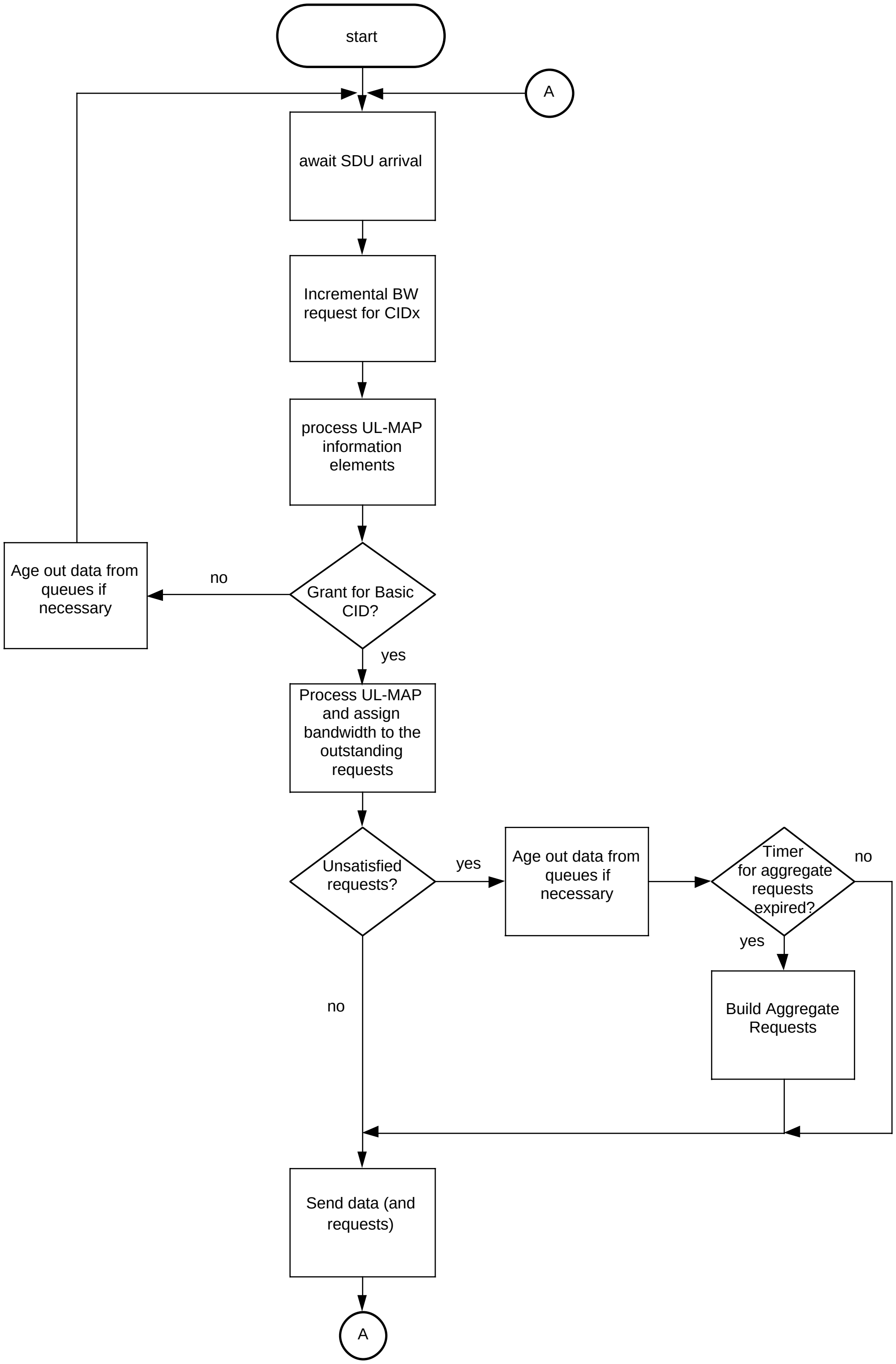
Polling is the process by which the BS allocates to the SSs bandwidth specifically for the purpose of making bandwidth requests. These allocations may be to individual SSs or to groups of SSs. Allocations to groups of connections and/or SSs actually define bandwidth request contention IEs. The allocations are not in the form of an explicit message, but are contained as a series of IEs within the uplink map.

Note that polling is done on either an SS or connection basis. Bandwidth is always requested on a CID basis and bandwidth is allocated on either a connection (GPC mode) or SS (GPSS mode) basis, based on the SS capability.

6.2.6.4.1 Unicast

When an SS is polled individually, no explicit message is transmitted to poll the SS. Rather, the SS is allocated, in the uplink map, bandwidth sufficient to respond with a bandwidth request. If the SS does not need bandwidth, it returns stuff bytes (0xFF). SSs operating in GPSS mode that have an active UGS connection of sufficient bandwidth shall not be polled individually unless they set the Poll Me (PM) bit in the header of a packet on the UGS connection. This saves bandwidth over polling all SSs individually. Note that unicast polling of a GPSS SS would normally be done on a per-SS basis by allocating a Data Grant IE directed at its Basic CID.

The information exchange sequence for individual polling is shown in Figure 34.


start
A
await SDU arrival
Incremental BW request for CIDx
process UL-MAP information elements
Grant for Basic CID?
no
Age out data from queues if necessary
yes
Process UL-MAP and assign bandwidth to the outstanding requests
Unsatisfied requests?
yes
Age out data from queues if necessary
Timer for aggregate requests expired?
no
yes
Build Aggregate Requests
no
Send data (and requests)
A

NOTE—The SS local scheduler decides which connections get the granted bandwidth.

Figure 33—SS GPSS mode flow chart

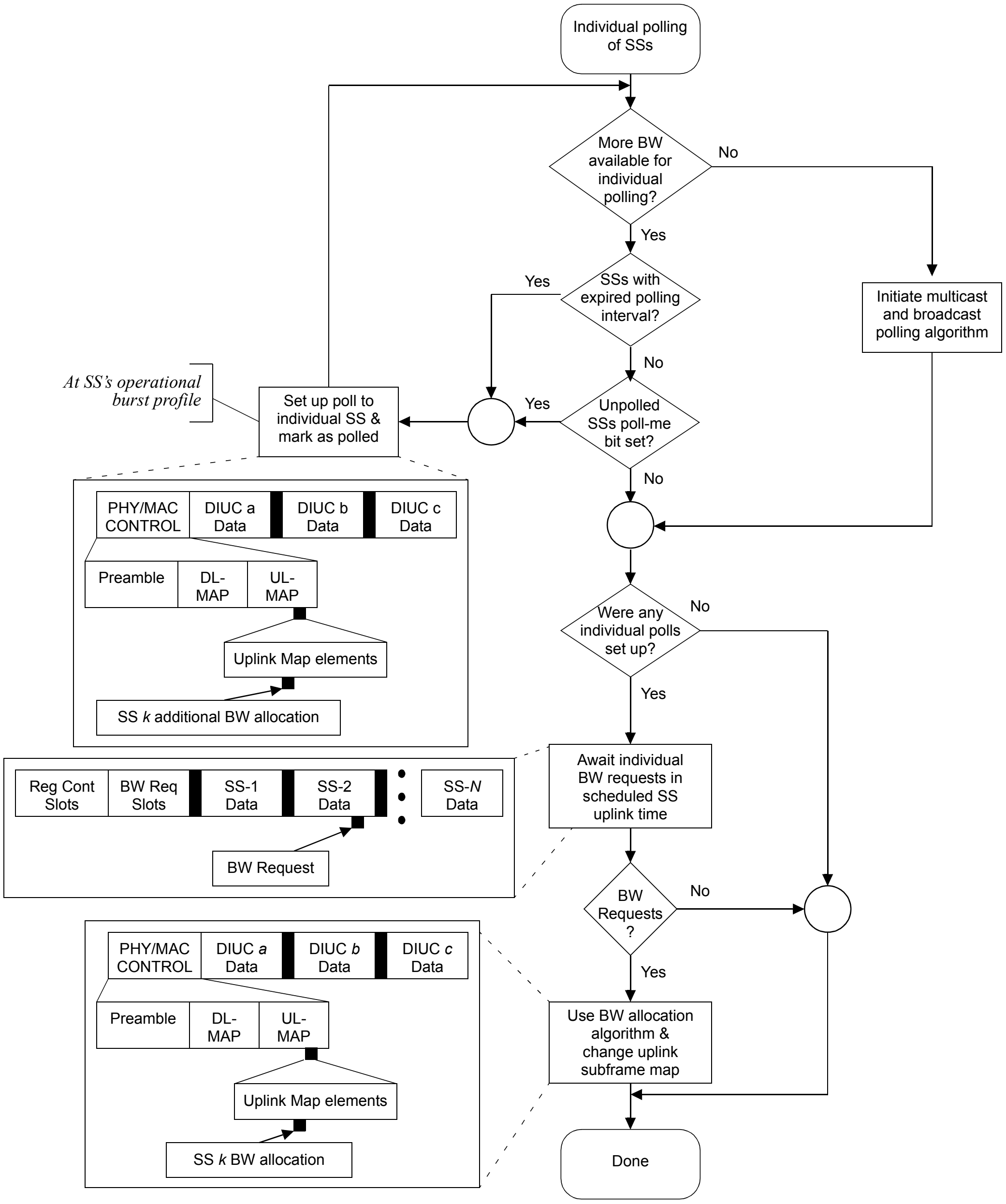

Individual polling of SSs
More BW available for individual polling?
No
Yes
SSs with expired polling interval?
Yes
No
Initiate multicast and broadcast polling algorithm
At SS's operational burst profile
Set up poll to individual SS & mark as polled
Yes
Unpolled SSs poll-me bit set?
No
PHY/MAC CONTROL
DIUC a Data
DIUC b Data
DIUC c Data
Preamble
DL-MAP
UL-MAP
Uplink Map elements
SS k additional BW allocation
Were any individual polls set up?
No
Yes
Reg Cont Slots
BW Req Slots
SS-1 Data
SS-2 Data
SS-N Data
BW Request
Await individual BW requests in scheduled SS uplink time
BW Requests ?
No
Yes
PHY/MAC CONTROL
DIUC a Data
DIUC b Data
DIUC c Data
Preamble
DL-MAP
UL-MAP
Uplink Map elements
SS k BW allocation
Use BW allocation algorithm & change uplink subframe map
Done

Figure 34—Unicast polling

6.2.6.4.2 Multicast and broadcast

If insufficient bandwidth is available to individually poll many inactive SSs, some SSs may be polled in multicast groups or a broadcast poll may be issued. Certain CIDs are reserved for multicast groups and for broadcast messages, as described in Table 121. As with individual polling, the poll is not an explicit message but bandwidth allocated in the uplink map. The difference is that, rather than associating allocated bandwidth with an SS's Basic CID, the allocation is to a multicast or broadcast CID. An example is provided in Table 59.

The information exchange sequence for multicast and broadcast polling is shown in Figure 35.

When the poll is directed at a multicast or broadcast CID, an SS belonging to the polled group may request bandwidth during any request interval allocated to that CID in the UL-MAP by a Request IE. In order to reduce the likelihood of collision with multicast and broadcast polling, only SS's needing bandwidth reply; they shall apply the contention resolution algorithm as defined in 6.2.8 to select the slot in which to transmit the initial bandwidth request. Zero-length bandwidth requests shall not be used in multicast or broadcast Request Intervals.

The SS shall assume that the transmission has been unsuccessful if no grant has been received in the number of subsequent UL-MAP messages specified by the parameter Random Access Timeout (see 11.1.1.1). Note that, with a frame-based PHY with UL-MAPs occurring at predetermined instants, erroneous UL-MAPs may be counted towards this number. If the rerequest is made in a multicast or broadcast opportunity, the SS continues to run the contention resolution algorithm in 6.2.8. Note that the SS is not restricted to issuing the rerequest in a multicast or broadcast Request Interval.

Table 59—Sample uplink map with multicast and broadcast IE

Interval description	Uplink map IE fields		
	CID (16 bits)	**UIUC (4 bits)**	**Offset (12 bits)**
Initial Ranging	0000	2	0
Multicast group 0xFFC5 Bandwidth Request	0xFFC5	1	405
Multicast group 0xFFDA Bandwidth Request	0xFFDA	1	605
Broadcast Bandwidth Request	0xFFFF	1	805
SS 5 Uplink Grant	0x007B	4	961
SS 21 Uplink Grant	0x01C9	7	1136
*	*	*	*
*	*	*	*
*	*	*	*

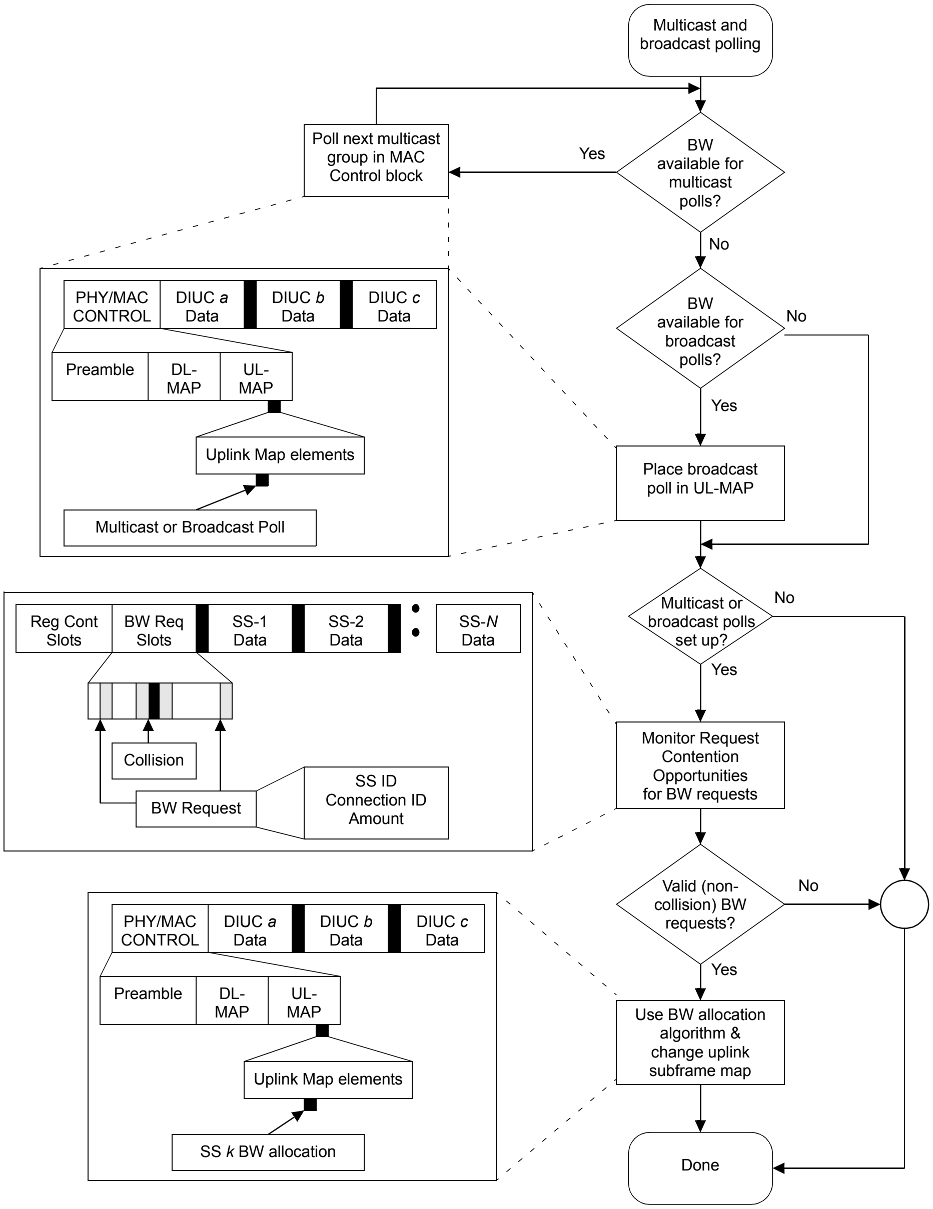

Multicast and broadcast polling
BW available for multicast polls?
Yes
Poll next multicast group in MAC Control block
No
BW available for broadcast polls?
No
Yes
Place broadcast poll in UL-MAP
Multicast or broadcast polls set up?
No
Yes
Monitor Request Contention Opportunities for BW requests
Valid (non-collision) BW requests?
No
Yes
Use BW allocation algorithm & change uplink subframe map
Done
PHY/MAC CONTROL
DIUC a Data
DIUC b Data
DIUC c Data
Preamble
DL-MAP
UL-MAP
Uplink Map elements
Multicast or Broadcast Poll
Reg Cont Slots
BW Req Slots
SS-1 Data
SS-2 Data
SS-N Data
Collision
BW Request
SS ID
Connection ID
Amount
PHY/MAC CONTROL
DIUC a Data
DIUC b Data
DIUC c Data
Preamble
DL-MAP
UL-MAP
Uplink Map elements
SS k BW allocation

Figure 35—Multicast and broadcast polling

6.2.6.4.3 Poll-me bit

SSs with currently active UGS connections may set the poll-me bit [bit PM in the Grant Management Sub-header (6.2.2.2.2)] in a MAC packet of the UGS connection to indicate to the BS that they need to be polled to request bandwidth for non-UGS connections. To reduce the bandwidth requirements of individual polling, SSs with active UGS connections need be individually polled only if the Poll-Me bit is set (or if the interval of the UGS is too long to satisfy the QoS of the SS's other connections). Once the BS detects this request for polling, the process for individual polling is used to satisfy the request. The procedure by which an SS stimulates the BS to poll it is shown in Figure 36. To minimize the risk of the BS missing the poll-me bit, the SS may set the bit in all UGS MAC Grant Management subheaders in the uplink scheduling interval.

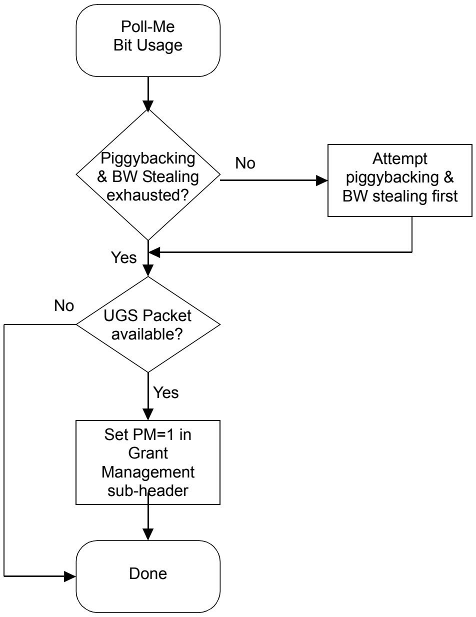

Poll-Me Bit Usage
Piggybacking & BW Stealing exhausted?
No
Attempt piggybacking & BW stealing first
Yes
No
UGS Packet available?
Yes
Set PM=1 in Grant Management sub-header
Done

Figure 36—Poll-Me bit usage

6.2.7 MAC support of PHY

Several duplexing techniques are supported by the MAC protocol. The choice of duplexing technique may affect certain PHY parameters as well as impact the features that can be supported.

The MAC is able to support both a framed and a nonframed PHY specification. For a framed PHY, the MAC aligns its scheduling intervals with the underlying PHY framing. For an unframed PHY, the scheduling intervals are chosen by the MAC to optimize system performance.

6.2.7.1 Unframed Frequency Division Duplexing (FDD)

In an Unframed FDD PHY, the uplink and downlink channels are located on separate frequencies and each subscriber station can transmit and receive simultaneously. In addition, uplink and downlink transmissions use no fixed duration frame. In this type of system, the downlink channel is “always on” and all subscriber stations are always listening to it. Therefore, traffic is sent in a broadcast manner using time division multiplexing (TDM) in the downlink channel. The uplink channel is shared using time division multiple access (TDMA), where a centralized scheduler controls the allocation of uplink bandwidth.

The current version of this standard does not define an Unframed FDD PHY.

6.2.7.2 Framed (burst) FDD

In a framed (burst) FDD system, the uplink and downlink channels are located on separate frequencies and the downlink data can be transmitted in bursts. A fixed duration frame is used for both uplink and downlink transmissions. This facilitates the use of different modulation types. It also allows simultaneous use of both full-duplex subscriber stations (which can transmit and receive simultaneously) and optionally half-duplex subscriber stations (which cannot). If half-duplex subscriber stations are used, the bandwidth controller shall not allocate uplink bandwidth for a half-duplex subscriber station at the same time that it is expected to receive data on the downlink channel.

Figure 37 describes the basics of the Framed FDD mode of operation. The fact that the uplink and downlink channels utilize a fixed duration frame simplifies the bandwidth allocation algorithms. A full-duplex subscriber station is capable of continuously listening to the downlink channel, while a half-duplex subscriber station can listen to the downlink channel only when it is not transmitting in the uplink channel.

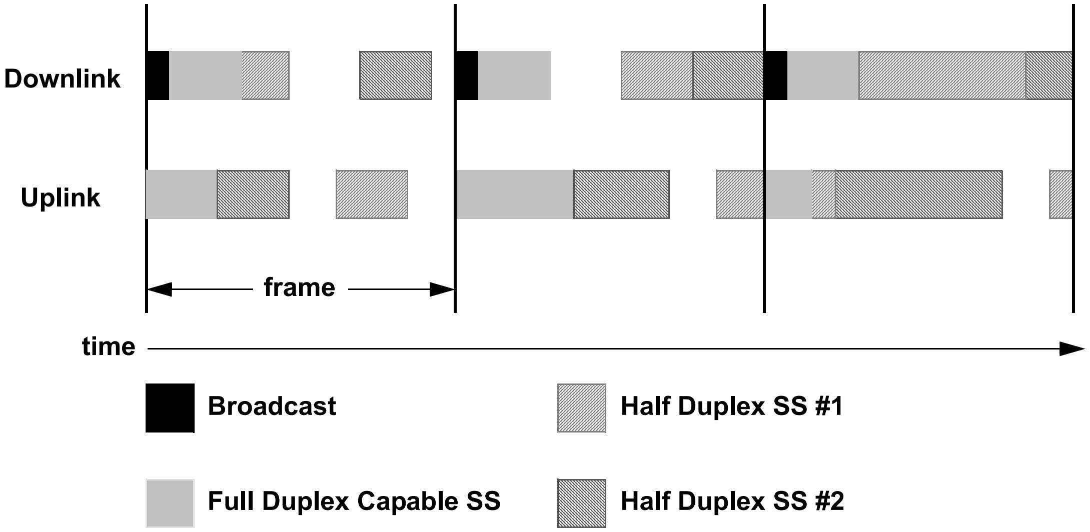


Figure 37—Example of Burst FDD bandwidth allocation

6.2.7.3 Time Division Duplexing (TDD)

In the case of TDD, the uplink and downlink transmissions occur at different times and usually share the same frequency. A TDD frame (see Figure 38) also has a fixed duration and contains one downlink and one uplink subframe. The frame is divided into an integer number of physical slots (PSs), which help to partition the bandwidth easily. The TDD framing is adaptive in that the bandwidth allocated to the downlink versus the uplink can vary. The split between uplink and downlink is a system parameter and is controlled at higher layers within the system.

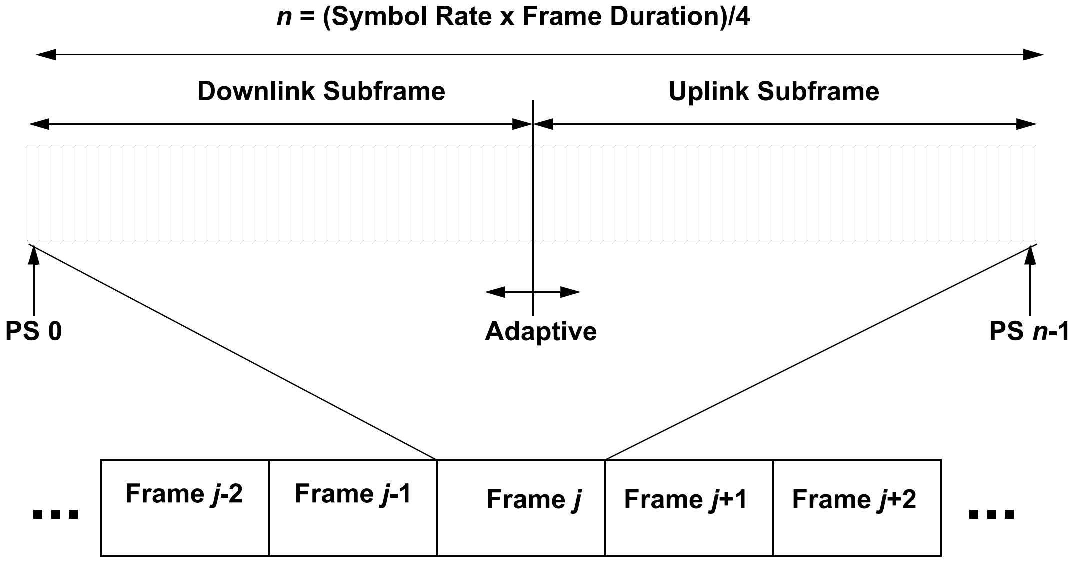


Figure 38—TDD frame structure

6.2.7.4 Downlink map

The Downlink Map (DL-MAP) message defines the usage of the downlink intervals for a burst mode physical layer.

6.2.7.5 Uplink map

The uplink map (UL-MAP) message defines the usage for the uplink minislots using a series of IEs, which define the usage of each uplink interval. The UL-MAP defines the uplink usage in terms of the offset from the previous IE start (the length) in numbers of minislots.

6.2.7.5.1 Uplink timing

The uplink timing is based on the Uplink Time Stamp reference, which is a counter that increments at a rate that is 16 times the PS rate. It therefore has a resolution that equals 1/16 of the PS duration. This allows the SS to track the BS clock with a small time offset.

6.2.7.5.1.1 Uplink timing with unframed PHY

In the case of an unframed PHY, the downlink map (DL-MAP) message broadcasts the Uplink Time Stamp value to all SSs. The Uplink Time Stamp from the BS is then used to adjust the SS's internal Time Stamp so that it tracks the BS timing. The SS Time Stamp is offset from the BS Time Stamp by the Timing Adjustment sent to each SS in the RNG-RSP message. The offset causes the uplink bursts to arrive at the BS at the proper time. After either the BS Time Stamp or SS Time Stamp reaches the maximum value of $2^{29} – 1$, the Time Stamp rolls over to zero and continues to count.

6.2.7.5.1.2 Uplink timing with framed PHY

In a framed PHY, there is no need for an Uplink Time Stamp counter as the uplink timing is referenced from the beginning of the downlink subframe. The Allocation Start Time in the UL-MAP is referenced from the start of the downlink subframe and may be such that the UL-MAP references some point in the current or next frame (see 6.2.7.6.1). The SS shall always adjust its concept of uplink timing based upon the Timing Adjustments sent in the RNG-RSP messages.

6.2.7.5.2 Uplink minislot definition

The uplink bandwidth allocation map (UL-MAP) uses units of minislots. The size of the minislot is specified as a number of PHY slots (PS) and is carried in the Uplink Channel Descriptor for each uplink channel. One mini-slot contains *n* PHY slots (PS), where *n* is an integer ranging from 0 through 255. There is no implication that any MAC PDU can actually be transmitted in a single minislot.

6.2.7.5.3 Uplink interval definition

All of the IEs defined below shall be supported by conformant SSs. Conformant BS may use any of these IEs when creating a UL-MAP message.

6.2.7.5.3.1 Request IE

Via the Request IE, the BS specifies an uplink interval in which requests may be made for bandwidth for uplink data transmission. The character of this IE changes depending on the type of CID used in the IE. If broadcast, this is an invitation for SSs to contend for requests. If unicast, this is an invitation for a particular SS to request bandwidth. Unicasts may be used as part of a QoS scheduling scheme that is vendor dependent. PDUs transmitted in this interval shall use the Bandwidth Request Header format (see 6.2.2).

6.2.7.5.3.2 Initial Maintenance IE

Via the Initial Maintenance IE, the BS specifies an interval in which new stations may join the network. A long interval, equivalent to the maximum round-trip propagation delay plus the transmission time of the RNG-REQ message, shall be provided in some UL-MAPs to allow new stations to perform initial ranging. Packets transmitted in this interval shall use the RNG-REQ MAC Management message format (see 6.2.2.3.5).

6.2.7.5.3.3 Station Maintenance IE

Via the Station Maintenance IE, the BS specifies an interval in which stations are expected to perform some aspect of routine network maintenance, such as ranging or power adjustment. The BS may request that a particular SS perform some task related to network maintenance, such as periodic transmit power adjustment. In this case, the Station Maintenance IE is unicast to provide uplink bandwidth in which to perform this task. Packets transmitted in this interval shall use the RNG-REQ MAC Management message format (see 6.2.2.3.5).

6.2.7.5.3.4 Data Grant Burst Type IEs

The Data Grant Burst Type IEs provide an opportunity for an SS to transmit one or more uplink PDUs. These IEs are issued either in response to a request from a station, or because of an administrative policy providing some amount of bandwidth to a particular station

There are six different Data Grant Burst Types that may be defined: Data Grant Burst Types 1 through 6 are associated with IUCs 4 through 9 respectively. Each Data Grant Burst Type description is defined in the UCD message.

6.2.7.5.3.5 Null IE

A Null IE terminates all actual allocations in the IE list. It is used to infer a length for the last interval.

6.2.7.5.3.6 Empty IE

The Empty IE indicates pauses in uplink transmissions. An SS shall not transmit during an Empty IE.

6.2.7.6 Map relevance and synchronization

6.2.7.6.1 Map relevance for framed PHY systems

The information in the DL-MAP pertains to the current frame (i.e., the frame in which it was received). The information carried in the UL-MAP pertains to a time interval starting at the Allocation Start Time measured from the beginning of the frame it was received in and ending after the last allocated minislot. This timing holds for both the TDD and FDD variants of the framed operation. The TDD variant is shown in Figure 39 and Figure 40. The FDD variant is shown in Figure 41 and Figure 42.

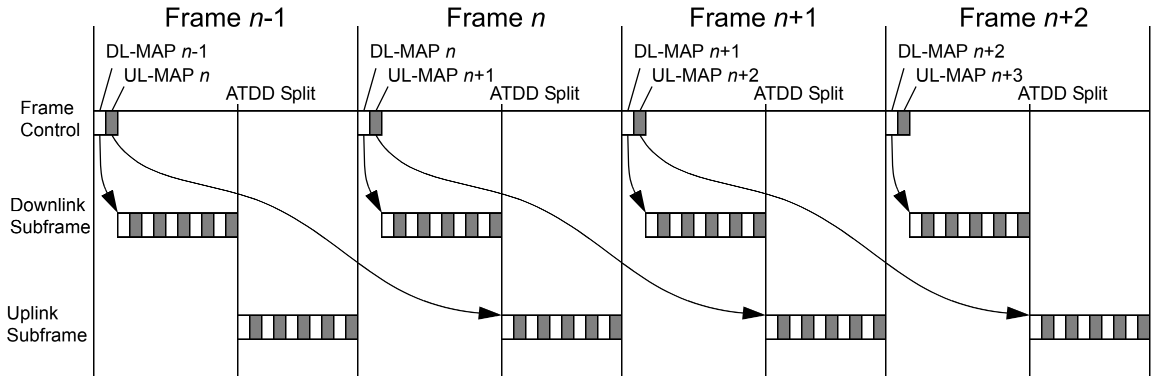


Figure 39—Maximum time relevance of PHY and MAC control information (TDD)

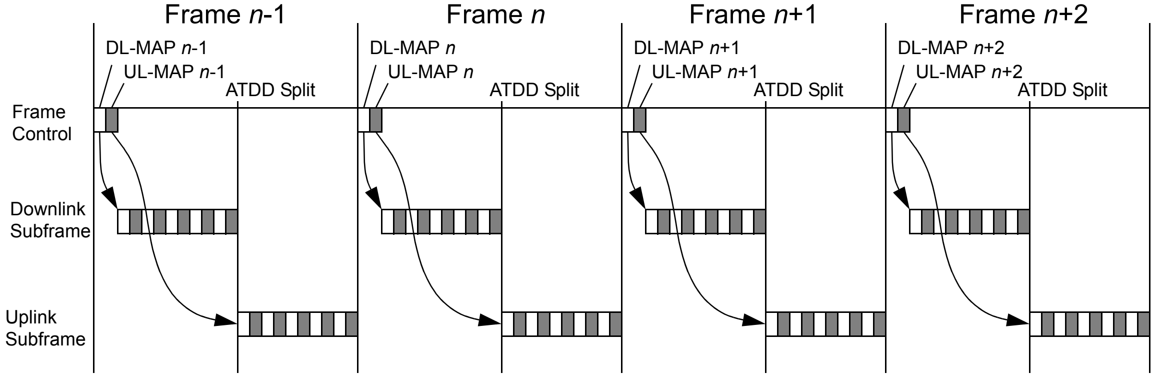


Figure 40—Minimum time relevance of PHY and MAC control information (TDD)

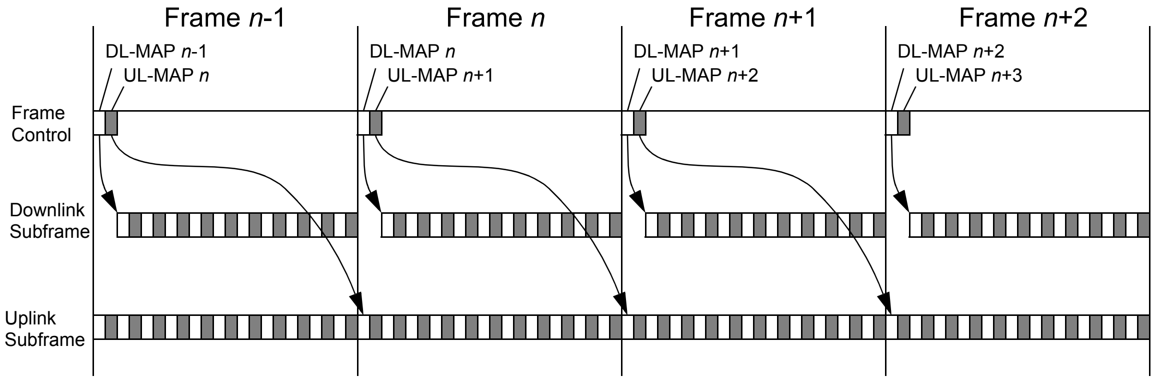


Figure 41—Maximum time relevance of PHY and MAC control information (FDD)

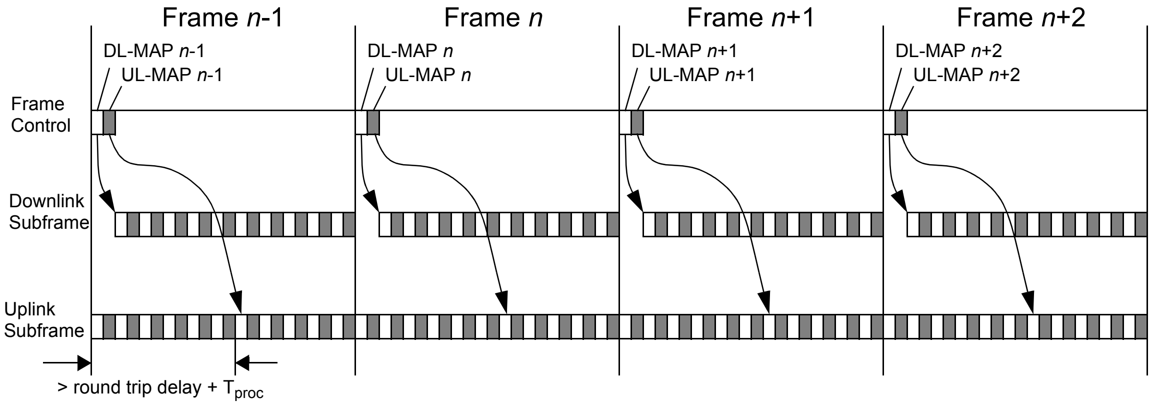


Figure 42—Minimum time relevance of PHY and MAC control information (FDD)

6.2.7.6.2 Map relevance for unframed PHY systems

In an unframed PHY system, the DL-MAP contains only an Uplink Time Stamp and does not define what information is being transmitted. All SSs continuously search the downlink signal for any downlink message that is addressed to them. The UL-MAP message in the downlink contains the Time Stamp that indicates the first minislot that the map defines.

The delay from the end of the UL-MAP to the beginning of the first Uplink interval defined by the map shall be greater than maximum round-trip delay plus the processing time required by the SS (see Figure 43).

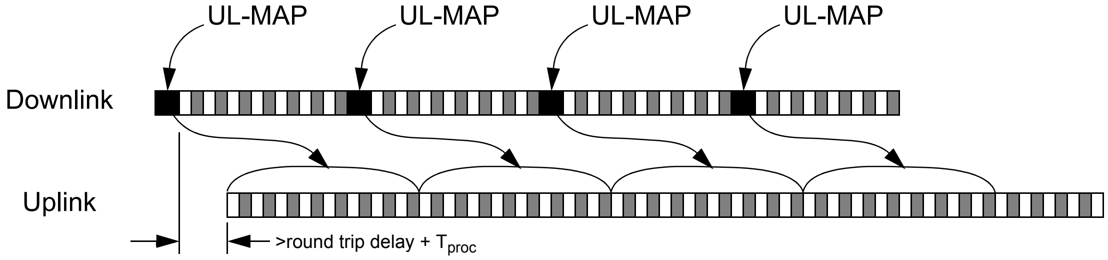


Figure 43—Time relevance of UL-Map information (unframed FDD)

6.2.8 Contention resolution

The BS controls assignments on the uplink channel through the UL-MAP messages and determines which minislots are subject to collisions. Collisions may occur during Initial Maintenance and Request intervals defined by their respective IEs. The potential occurrence of collisions in Request Intervals is dependent on the CID in the respective IE. This subclause describes uplink transmission and contention resolution. For simplicity, it refers to the decisions an SS makes. Since an SS can have multiple uplink service flows (each with its own CID), it makes these decisions on a per CID or per service QoS basis.

The mandatory method of contention resolution which shall be supported is based on a truncated binary exponential backoff, with the initial backoff window and the maximum backoff window controlled by the BS. The values are specified as part of the UCD message and represent a power-of-two value. For example, a value of 4 indicates a window between 0 and 15; a value of 10 indicates a window between 0 and 1023.

When an SS has information to send and wants to enter the contention resolution process, it sets its internal backoff window equal to the Request (or Ranging for initial ranging) Backoff Start defined in the UCD message referenced by the UCD Count in the UL-MAP message currently in effect.[11]

The SS shall randomly select a number within its backoff window. This random value indicates the number of contention transmission opportunities that the SS shall defer before transmitting. An SS shall consider only contention transmission opportunities for which this transmission would have been eligible. These are defined by Request IEs (or Initial Maintenance IEs for initial ranging) in the UL-MAP messages. Note that each IE may consist of multiple contention transmission opportunities.

Using bandwidth requests as an example, consider an SS whose initial backoff window is 0 to 15 and assume it randomly selects the number 11. The SS must defer a total of 11 contention transmission opportunities. If the first available Request IE is for 6 requests, the SS does not use this and has 5 more opportunities to defer. If the next Request IE is for 2 requests, the SS has 3 more to defer. If the third Request IE is for 8 requests, the SS transmits on the fourth opportunity, after deferring for 3 more opportunities.

After a contention transmission, the SS waits for a Data Grant Burst Type IE in a subsequent map (or waits for a RNG-RSP message for initial ranging). Once received, the contention resolution is complete.

The SS shall consider the contention transmission lost if no data grant has been given within T16 (or no response within T3 for initial ranging). The SS shall now increase its backoff window by a factor of two, as long as it is less than the maximum backoff window. The SS shall randomly select a number within its new backoff window and repeat the deferring process described above.

[11] The map currently in effect is the map whose allocation start time has occurred but which includes IEs that have not occurred.

This retry process continues until the maximum number (i.e., Request Retries for bandwidth requests and Contention Ranging Retries for initial ranging) of retries has been reached. At this time, for bandwidth requests, the PDU shall be discarded. For initial ranging, proper actions are specified in 6.2.9.5. Note that the maximum number of retries is independent of the initial and maximum backoff windows that are defined by the BS.

For bandwidth requests, if the SS receives a unicast Request IE or Data Grant Burst Type IE at any time while deferring for this CID, it shall stop the contention resolution process and use the explicit transmission opportunity.

The BS has much flexibility in controlling the contention resolution. At one extreme, the BS may choose to set up the Request (or Ranging) Backoff Start and Request (or Ranging) Backoff End to emulate an Ethernet-style backoff with its associated simplicity and distributed nature as well as its fairness and efficiency issues. This would be done by setting Request (or Ranging) Backoff Start = 0 and Request (or Ranging) Backoff End = 10 in the UCD message. At the other end, the BS may make the Request (or Ranging) Backoff Start and Request (or Ranging) Backoff End identical and frequently update these values in the UCD message so that all SS are using the same, and hopefully optimal, backoff window.

6.2.8.1 Transmission opportunities

A transmission opportunity is defined as any minislot in which an SS is allowed to start a transmission. The number of transmission opportunities associated with a particular IE in a map is dependent on the total size of the interval as well as the size of an individual transmission.

As an example, consider contention-based bandwidth requests for a system where the PHY protocol has a frame duration of 1 ms, 4 symbols for each PS, 2 PSs for each minislot, an uplink preamble of 16 symbols (i.e., 2 minislots), and an SS Transition Gap of 24 symbols (i.e., 3 minislots). Thus, assuming QPSK modulation, each transmission opportunity requires 8 minislots: 3 for the SS Transition Gap, 2 for the preamble, and 3 for the bandwidth request message.

If the BS schedules a Request IE of, for example, 24 minislots, there will be three transmission opportunities within this IE. Details of the three transmission opportunities are shown in Figure 44.

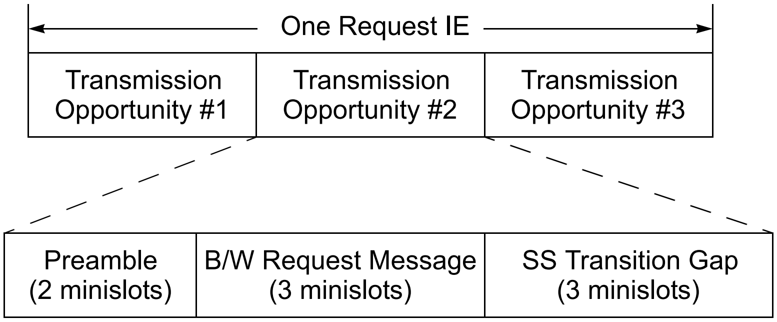

One Request IE
Transmission Opportunity #1
Transmission Opportunity #2
Transmission Opportunity #3
Preamble (2 minislots)
B/W Request Message (3 minislots)
SS Transition Gap (3 minislots)

Figure 44—Example of Request IE containing multiple transmission opportunities

6.2.9 Network entry and initialization

The procedure for initialization of an SS shall be as shown in Figure 45. This figure shows the overall flow between the stages of initialization in an SS. This shows no error paths and is shown simply to provide an overview of the process. The more detailed finite state machine representations of the individual sections (including error paths) are shown in the subsequent figures. Timeout values are defined in 10.1.

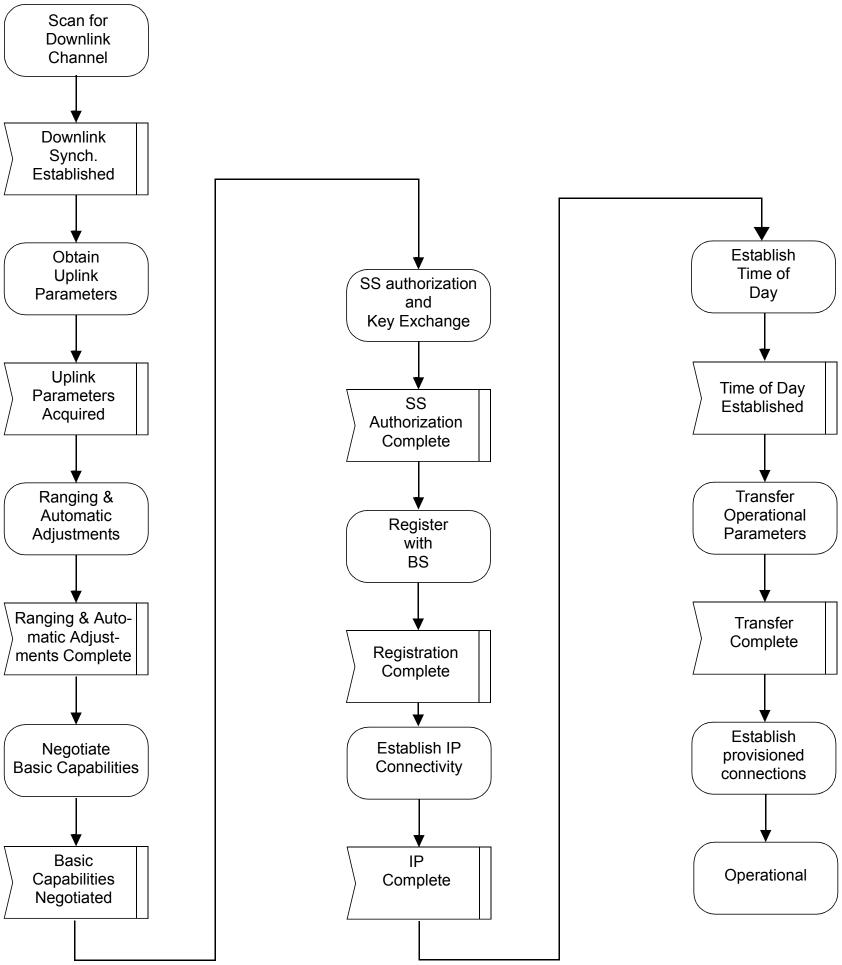

Scan for Downlink Channel
Downlink Synch. Established
Obtain Uplink Parameters
Uplink Parameters Acquired
Ranging & Automatic Adjustments
Ranging & Automatic Adjustments Complete
Negotiate Basic Capabilities
Basic Capabilities Negotiated
SS authorization and Key Exchange
SS Authorization Complete
Register with BS
Registration Complete
Establish IP Connectivity
IP Complete
Establish Time of Day
Time of Day Established
Transfer Operational Parameters
Transfer Complete
Establish provisioned connections
Operational

Figure 45—SS Initialization overview

The procedure can be divided into the following phases:Scan for downlink channel and establish synchronization with the BS

c) Obtain transmit parameters (from UCD message)

d) Perform ranging

e) Negotiate basic capabilities

f) Authorize SS and perform key exchange

g) Perform registration

h) Establish IP connectivity

i) Establish time of day

j) Transfer operational parameters

k) Set up connections

Each SS contains the following information when shipped from the manufacturer:

a) A 48-bit universal MAC address (per IEEE Std 802-2001) assigned during the manufacturing process. This is used to identify the SS to the various provisioning servers during initialization.

b) Security information as defined in Clause 7 (e.g., X.509 certificate) used to authenticate the SS to the security server and authenticate the responses from the security and provisioning servers.

6.2.9.1 Scanning and synchronization to the downlink

On initialization or after signal loss, the SS shall acquire a downlink channel. The SS shall have nonvolatile storage in which the last operational parameters are stored and shall first try to reacquire this downlink channel. If this fails, it shall begin to continuously scan the possible channels of the downlink frequency band of operation until it finds a valid downlink signal.

Once the PHY has achieved synchronization, as given by a PHY Indication, the MAC Sublayer shall attempt to acquire the channel control parameters for the downlink and then the uplink.

6.2.9.2 Obtain downlink parameters

The MAC sublayer shall search for the DL-MAP MAC management messages. The SS achieves MAC synchronization once it has received at least one DL-MAP message. An SS MAC remains in synchronization as long as it continues to successfully receive the DL-MAP and DCD messages for its Channel. If the Lost DL-MAP Interval (Table 118) has elapsed without a valid DL-MAP message or the T1 interval (Table 118) has elapsed without a valid DCD message, an SS shall try to reestablish synchronization. The process of acquiring synchronization is illustrated in Figure 46. The process of maintaining synchronization is illustrated in Figure 47.

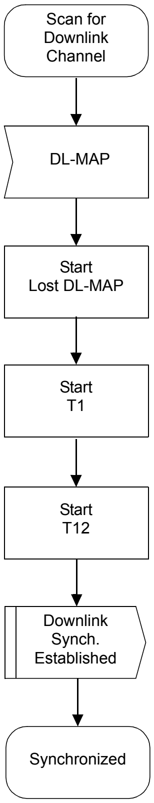


Figure 46—Obtaining downlink synchronization

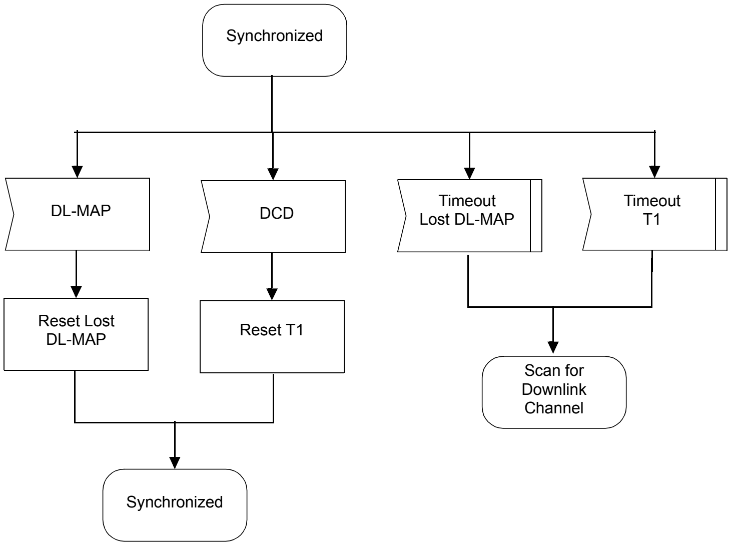


Figure 47—Maintaining downlink synchronization

6.2.9.3 Obtain uplink parameters

After synchronization, the SS shall wait for UCD message from the BS in order to retrieve a set of transmission parameters for a possible uplink channel. These messages are transmitted periodically from the BS for all available uplink channels and are addressed to the MAC broadcast address. The SS shall determine whether it may use the uplink channel from the channel description parameters.

The SS shall collect all UCDs which are different in their channel ID field to build a set of usable channel IDs. If no channel can be found after a suitable timeout period, then the SS shall continue scanning to find another downlink channel. The process of obtaining uplink parameters is illustrated in Figure 48.

The SS shall determine from the channel description parameters whether it may use the uplink channel. If the channel is not suitable, then the SS shall try the next channel ID until it finds a usable channel. If the channel is suitable, the SS shall extract the parameters for this uplink from the UCD. It then shall wait for the next DL-MAP message and extract the time synchronization from this message. The SS then shall wait for a bandwidth allocation map for the selected channel. It may begin transmitting uplink in accordance with the MAC operation and the bandwidth allocation mechanism.

The SS shall perform initial ranging at least once, per Figure 50. If initial ranging is not successful, then the next channel ID is selected, and the procedure restarted from UCD extraction. When there are no more channel IDs to try, then the SS shall continue scanning to find another downlink channel.

The SS MAC is considered to have valid uplink parameters as long as it continues to successfully receive the UL-MAP and UCD messages. If at least one of these messages is not received within the time intervals specified in Table 118, the SS shall not use the uplink. This is illustrated in Figure 49.

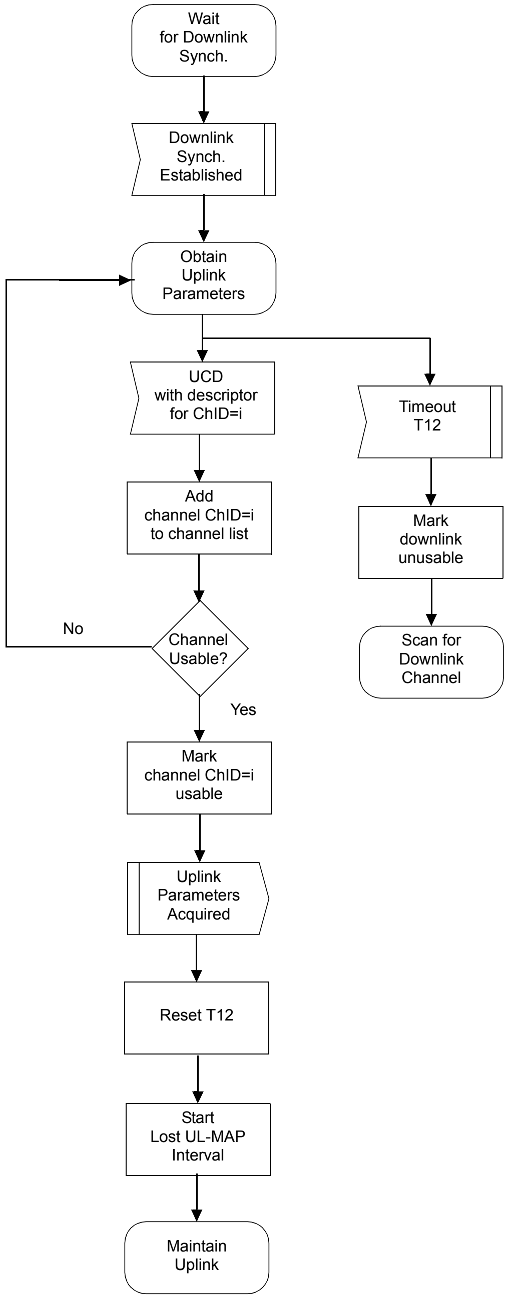

Wait for Downlink Synch.
Downlink Synch. Established
Obtain Uplink Parameters
UCD with descriptor for ChID=i
Timeout T12
Add channel ChID=i to channel list
Mark downlink unusable
No
Channel Usable?
Scan for Downlink Channel
Yes
Mark channel ChID=i usable
Uplink Parameters Acquired
Reset T12
Start Lost UL-MAP Interval
Maintain Uplink

Figure 48—Obtaining uplink parameters

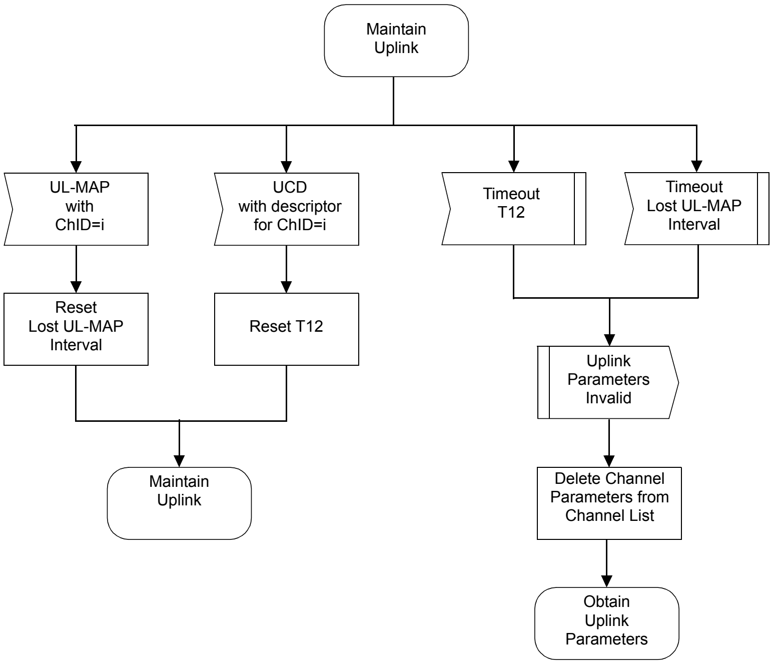

Maintain Uplink
UL-MAP with ChID=i
UCD with descriptor for ChID=i
Timeout T12
Timeout Lost UL-MAP Interval
Reset Lost UL-MAP Interval
Reset T12
Uplink Parameters Invalid
Maintain Uplink
Delete Channel Parameters from Channel List
Obtain Uplink Parameters

Figure 49—Maintain uplink parameters, single uplink case, ChID=i

6.2.9.4 Message flows during scanning and uplink parameter acquisition

The BS shall generate UCD and DCD messages on the downlink at periodic intervals within the ranges defined in Table 118. The BS shall generate UL-MAP and DL-MAP at intervals as specified in a particular PHY specification. These messages are addressed to all SSs. Refer to Table 60.

Table 60—Message flows during scanning and uplink parameter acquisition

BS		SS
clock time to send DL-MAP	----------------DL-MAP---------------->	|
clock time to send UCD and DCD	--------------UCD and DCD--------------->	|
		|
clock time to send DL-MAP	----------------DL-MAP---------------->	|
		| Example of a UCD and DCD
		| cycle prior to SS power-on
		|
clock time to send DL-MAP	----------------DL-MAP---------------->	|
		|
clock time to send DL-MAP	----------------DL-MAP---------------->	|
clock time to send DL-MAP	----------------DL-MAP---------------->	
clock time to send UCD and DCD	----------------UCD and DCD------------>	
clock time to send DL-MAP	----------------DL-MAP---------------->	

Table 60—Message flows during scanning and uplink parameter acquisition *(continued)*

BS		SS
		power on sequence complete
clock time to send DL-MAP	----------------DL-MAP---------------->	
clock time to send DCD	-------------------DCD------------------>	
		establish PHY synchronization & wait for UCD
clock time to send DL-MAP	----------------DL-MAP----------------->	
clock time to send DL-MAP	----------------DL-MAP------------------>	
clock time to send UCD	----------------UCD------------------------>	
		obtain parameters for this uplink channel to use for initialization
clock time to send DL-MAP	----------------DL-MAP------------------->	
		extract slot info for uplink & wait for transmission opportunity to perform ranging
clock time to send DL-MAP	----------------DL-MAP------------------->	
clock time to send UL-MAP	----------------UL-MAP------------------->	
		start ranging process

6.2.9.5 Initial ranging and automatic adjustments

Ranging is the process of acquiring the correct timing offset such that the SS's transmissions are aligned to a symbol that marks the beginning of a minislot boundary. The timing delays through the PHY layer shall be relatively constant. Any variation in the PHY delays shall be accounted for in the guard time of the uplink PHY overhead.

First, an SS shall synchronize to the downlink and learn the uplink channel characteristics through the UCD MAC management message. At this point, the SS shall scan the UL-MAP message to find an Initial Maintenance Interval. The BS shall make an Initial Maintenance Interval large enough to account for the variation in delays between any two SSs (maximum round-trip propagation delay due to cell radius plus maximum allowable implementation delay).

The SS shall put together a RNG-REQ message to be sent in an Initial Maintenance Interval. The CID field shall be set to the noninitialized SS value (zero).

Ranging adjusts each SS's timing offset such that it appears to be colocated with the BS. The SS shall set its initial timing offset to the amount of internal fixed delay equivalent to colocating the SS next to the BS. This amount includes delays introduced through a particular implementation and shall include the downlink PHY interleaving latency, if any.

When the Initial Maintenance transmission opportunity occurs, the SS shall send the RNG-REQ message. Thus, the SS sends the message as if it were colocated with the BS.

The SS shall first send the RNG-REQ at minimum power level, and if it is not successful, the SS shall resend it at the next Initial Maintenance transmission opportunity at one step higher power level until successful.

Once the BS has successfully received the RNG-REQ message, it shall return a RNG-RSP message addressed to the individual SS. Within the RNG-RSP message shall be the Basic and Primary Management CIDs assigned to this SS. The message shall also contain information on RF power level adjustment and offset frequency adjustment as well as any timing offset corrections.

The SS shall now wait for an individual Station Maintenance region assigned to its Basic CID. It shall now transmit another RNG-REQ message at this time using the Basic CID along with any power level and timing offset corrections.

The BS shall return another RNG-RSP message to the SS with any additional fine tuning required. The ranging request/response steps shall be repeated until the response contains a Ranging Successful notification or the BS aborts ranging. Once successfully ranged (RNG-REQ is within tolerance of the BS), the SS shall join normal data traffic in the uplink. In particular, state machines and the applicability of retry counts and timer values for the ranging process are defined in Table 118.

NOTE—The burst profile to use for any uplink transmission is defined by the Uplink Interval Usage Code (UIUC). Each UIUC is mapped to a burst profile in the UCD message.

The message sequence chart (Table 61) and flow charts (Figure 50 and Figure 51) on the following pages define the ranging and adjustment process which shall be followed by compliant SSs and BSs.

Table 61—Ranging and automatic adjustments procedure

BS		SS
[time to send the Initial Maintenance Interval]		
send map containing Initial Maintenance information element with a broadcast CID	-----------UL-MAP------------>	
	<---------RNG-REQ-------	transmit ranging packet in contention mode with CID parameter = 0
[receive recognizable ranging packet]		
allocate Basic and Primary Management CID		
send ranging response	----------RNG-RSP------->	
add Basic CID to poll list		recognize own MAC Address, store Basic and Primary Management CID, and adjust other parameters
[time to send the next map]		

Table 61—Ranging and automatic adjustments procedure *(continued)*

BS		SS
send map with Station Maintenance information element to SS using Basic CID	-----------UL-MAP----------->	recognize own Basic CID in map
	<---------RNG-REQ-------	reply to Station Maintenance opportunity poll
send ranging response	----------RNG-RSP------->	
		adjust local parameters
send periodic transmission opportunity to broadcast address	------------UL-MAP---------->	

NOTES:

1—The BS shall allow the SS sufficient time to have processed the previous RNG-RSP (i.e., to modify the transmitter parameters) before sending the SS a specific ranging opportunity. This is defined as SS Ranging Response Processing Time in Table 118.

2—For multichannel support, the SS shall attempt initial ranging on every suitable uplink channel before moving to the next available downlink channel.

On receiving a RNG-RSP instruction to move to a new downlink frequency and/or uplink channel ID, the SS shall consider any previously assigned Basic, Primary Management, and Secondary Management CIDs to be deassigned, and shall obtain new Basic, Primary Management, and Secondary Management CIDs via initial ranging and registration.

It is possible that the RNG-RSP may be lost after transmission by the BS. The SS shall recover by timing out and reissuing its Initial RNG-REQ. Since the SS is uniquely identified by the source MAC address in the Ranging Request, the BS may immediately reuse the Basic, Primary Management, and Secondary Management CIDs previously assigned. If the BS assigns new Basic, Primary Management, and Secondary Management CIDs, it shall make some provision for aging out the old CIDs that went unused.

6.2.9.6 Ranging parameter adjustment

Adjustment of local parameters (e.g., transmit power) in an SS as a result of the receipt (or nonreceipt) of a RNG-RSP is considered to be implementation-dependent with the following restrictions:

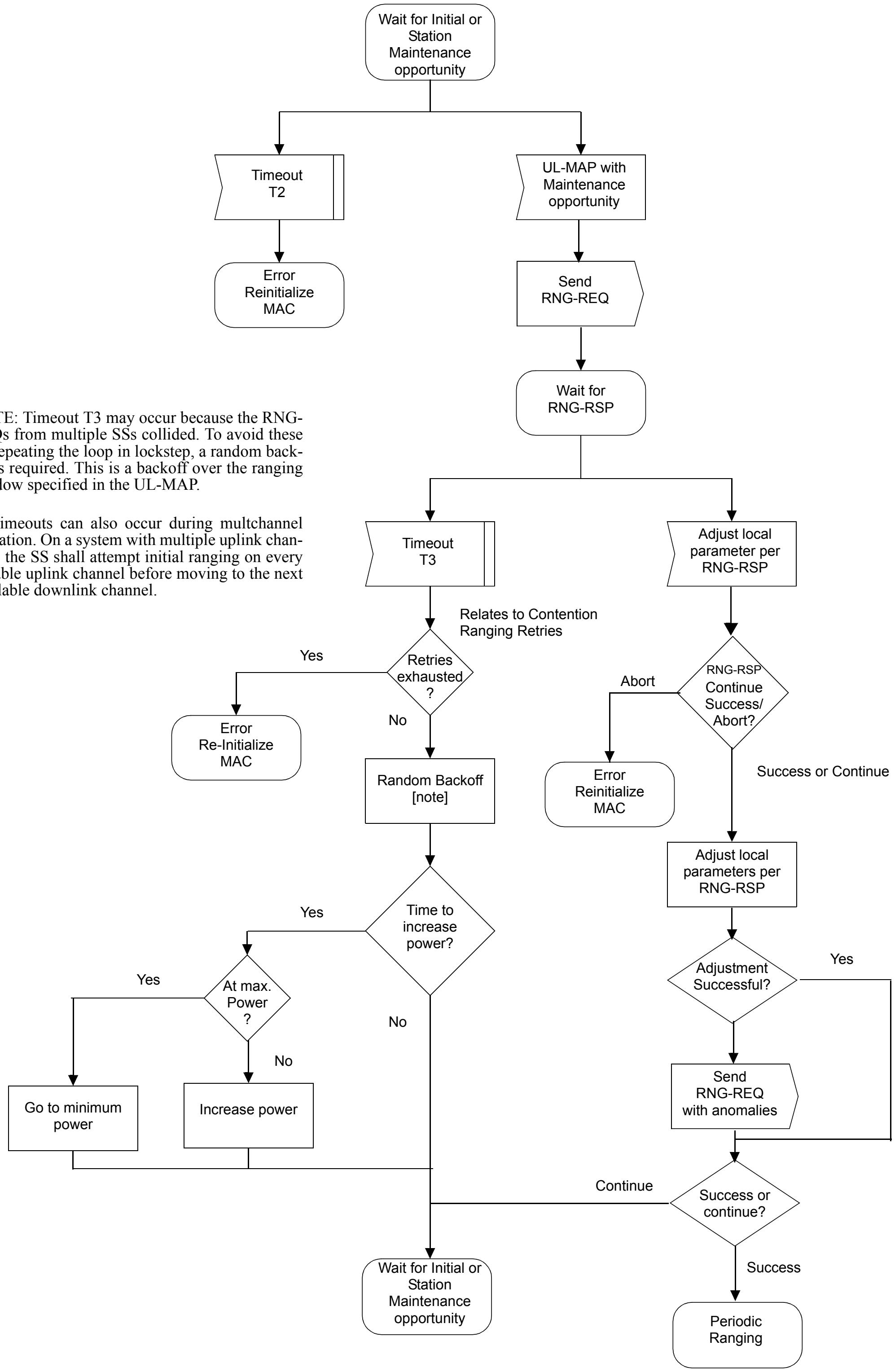
a) All parameters shall be within the approved range at all times.

b) Power adjustment shall start from the minimum value unless a valid power is available from non-volatile storage, in which case this shall be used as a starting point.

c) Power adjustment shall be capable of being reduced or increased by the specified amount in response to RNG-RSP messages.

d) If, during initialization, power is increased to the maximum value (without a response from the BS) it shall wrap back to the minimum

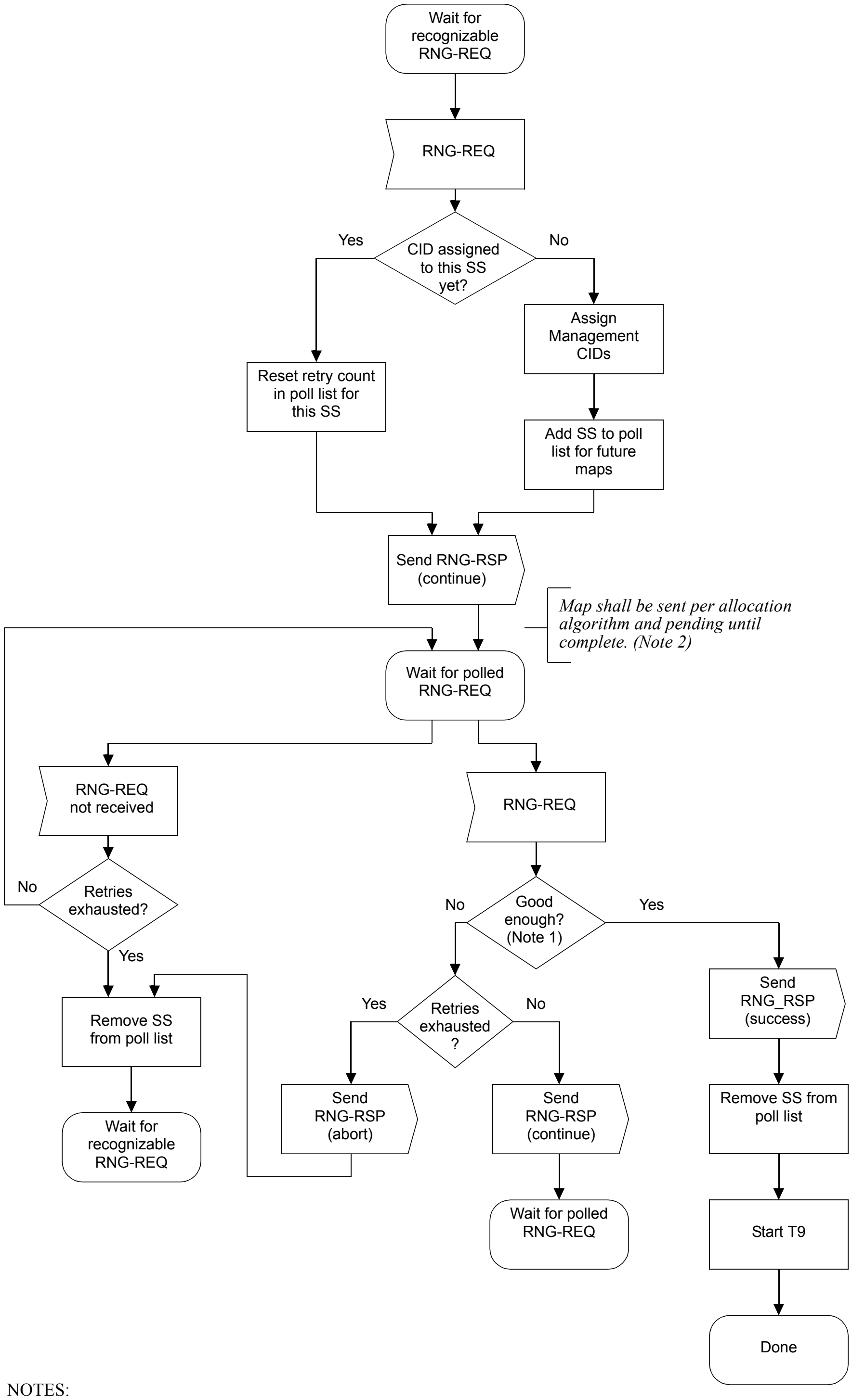
On receiving a RNG-RSP, the SS shall not transmit until the RF signal has been adjusted in accordance with the RNG-RSP and has stabilized.



NOTE: Timeout T3 may occur because the RNG-REQs from multiple SSs collided. To avoid these SS repeating the loop in lockstep, a random backoff is required. This is a backoff over the ranging window specified in the UL-MAP.

T3 timeouts can also occur during multchannel operation. On a system with multiple uplink channels, the SS shall attempt initial ranging on every suitable uplink channel before moving to the next available downlink channel.

Figure 50—Initial Ranging—SS



NOTES:

1—Means ranging is within the tolerable limits of the BS.

2—RNG-REQ pending-until-complete was nonzero, the BS should hold off the station maintenance opportunity accordingly unless needed, for example, to adjust the SS's power level.

3—"Retries exhausted?" conditional refers to the "Invited Ranging Retries" entry of Table 118.

Figure 51—Initial Ranging—BS

6.2.9.7 Negotiate basic capabilities

Immediately after completion of ranging, the SS informs the BS of its basic capabilities by transmitting an SBC-REQ message with its capabilities set to “on.” The BS responds with an SBC-RSP message with the intersection of the SS’s and the BS’s capabilities set to “on.”

6.2.9.8 SS authorization and key exchange

The BS and SS shall perform authorization and key exchange as described in 7.2.

6.2.9.9 Registration

Registration is the process by which the SS receives its Secondary Management CID and thus becomes manageable. To register with a BS, the SS shall send a REG-REQ message to the BS. The BS shall respond with a REG-RSP message. The REG-RSP message shall include the Secondary Management CID.

Figure 52 shows the procedure that shall be followed by the SS.

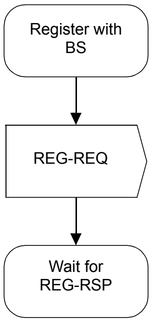


Figure 52—Registration—SS

Once the SS has sent a REG-REQ to the BS, it shall wait for a REG-RSP to authorize it to forward traffic to the network. Figure 53 shows the waiting procedure that shall be followed by the SS.

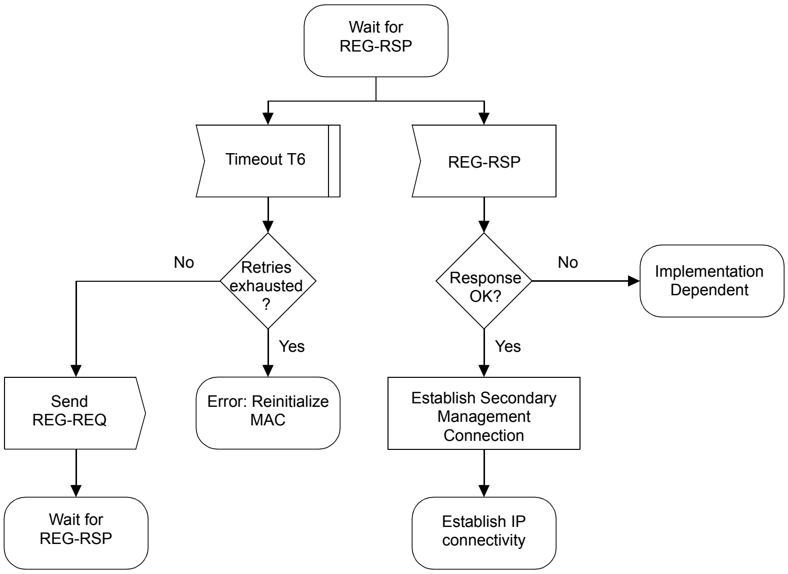


Figure 53—Wait for registration response—SS

The BS shall perform the operations shown in Figure 54.

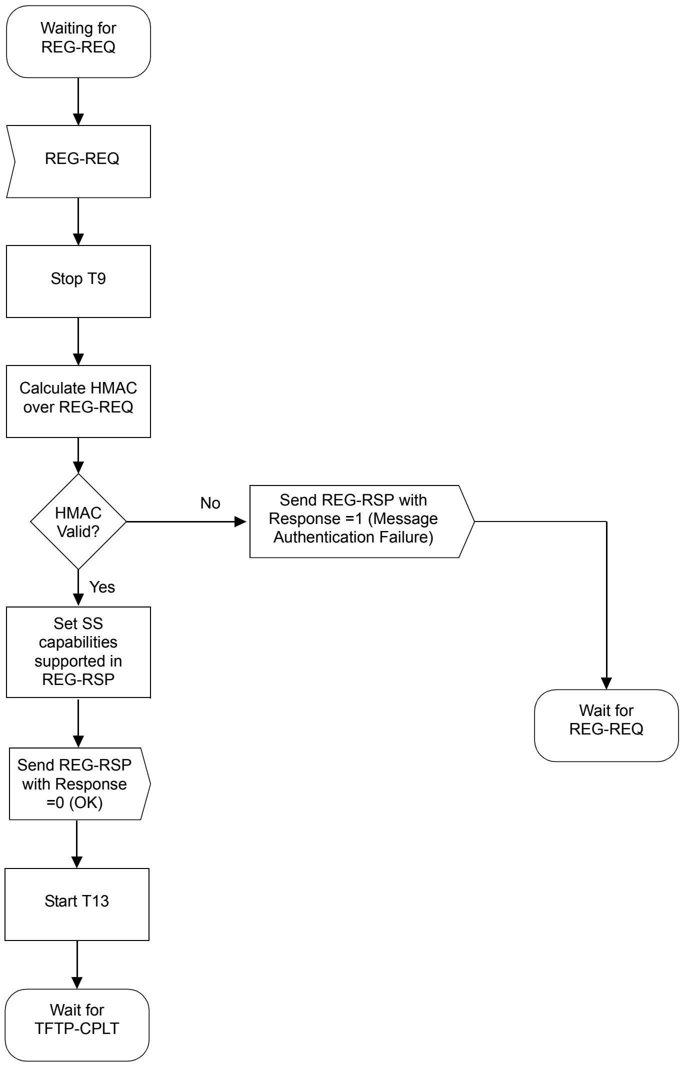

Waiting for REG-REQ
REG-REQ
Stop T9
Calculate HMAC over REG-REQ
HMAC Valid?
No
Send REG-RSP with Response =1 (Message Authentication Failure)
Wait for REG-REQ
Yes
Set SS capabilities supported in REG-RSP
Send REG-RSP with Response =0 (OK)
Start T13
Wait for TFTP-CPLT

Figure 54—Registration—BS

Upon sending a REG-RSP, the BS shall wait for a TFTP-CPLT. If timer T9 (defined in Table 118) expires, the BS shall both de-assign the management CIDs from that SS and make some provision for aging out those CIDs.

6.2.9.9.1 IP version negotiation

The SS may include the IP Version (11.4.1.7) parameter in the REG-REQ to indicate which versions of IP it supports on the Secondary Management Connection. When present in the REG-REQ, the BS shall include the IP Version parameter (11.4.1.7) in the REG-RSP to command the SS to use the indicated version of IP on the Secondary Management Connection. The BS shall command the use of exactly one of the IP versions supported by the SS.

The omission of the IP Version parameter in the REG-REQ shall be interpreted as IPv4 support only. Consequently, omission of the IP Version parameter in the REG-RSP shall be interpreted as a command to use IPv4 on the Secondary Management Connection.

6.2.9.10 Establish IP connectivity

At this point, the SS shall invoke Dynamic Host Configuration Protocol (DHCP) mechanisms [IETF RFC 2131] in order to obtain an IP address and any other parameters needed to establish IP connectivity. The DHCP response shall contain the name of a file which contains further configuration parameters. Establishment of IP connectivity shall be performed on the SS's Secondary Management Connection; see Table 62.

Table 62—Establishing IP connectivity

SS		DHCP
send DHCP request to broadcast address		
	----------------DHCP discover------------>	
		check SS MAC address & respond
	<--------------DHCP offer -----------------	
choose server		
	----------------DHCP request-------------->	
		process request
	<--------------DHCP response--------------	
set up IP parameters from DHCP response		

6.2.9.11 Establish time of day

The SS and BS need to have the current date and time. This is required for time-stamping logged events for retrieval by the management system. This need not be authenticated and need be accurate only to the nearest second.

The protocol by which the time of day shall be retrieved is defined in [IETF RFC 868]. Refer to Table 63. The request and response shall be transferred using user datagram protocol (UDP). The time retrieved from the server [Universal Coordinated Time (UTC)] shall be combined with the time offset received from the DHCP response to create the current local time. Establishment of time of day shall be performed on the SS's Secondary Management Connection.

Table 63—Establishing time of day

SS		Time Server
send request to time server		
	----------------time of day request------------>	
		process request
	<--------------time of day response--------------	
set up / correct time of day from response		

Successfully acquiring the Time of Day is not mandatory for a successful registration, but is necessary for ongoing operation. The specific timeout for Time of Day Requests is implementation dependent. However, the SS shall not exceed more than 3 Time of Day requests in any 5 min period.

6.2.9.12 Transfer operational parameters

After DHCP is successful, the SS shall download the SS Configuration File (9.2) using TFTP on the SS's Secondary Management Connection, as shown in Figure 52. The TFTP Configuration File server is specified by the "siaddr" field of the DHCP response. The SS shall use an adaptive timeout for TFTP based on binary exponential backoff [IETF RFC 1123, IETF RFC 2349].

The parameter fields required in the DHCP response and the format and content of the configuration file shall be as defined in 9.2. Note that these fields are the minimum required for interoperability.

6.2.9.13 Establish provisioned connections

After the transfer of operational parameters, the BS shall send DSA-REQ messages to the BS to set up connections for preprovisioned service flows belonging to the SS. The SS responds with DSA-RSP messages. This is described further in 6.2.13.7.1.

6.2.10 Ranging

The BS shall provide each SS a periodic Ranging opportunity at an interval sufficiently shorter than T4 that a map could be missed without the SS timing out. The size of this "subinterval" is BS dependent. For GPSS mode subscriber stations, any allocation of uplink bandwidth constitutes a Ranging opportunity.

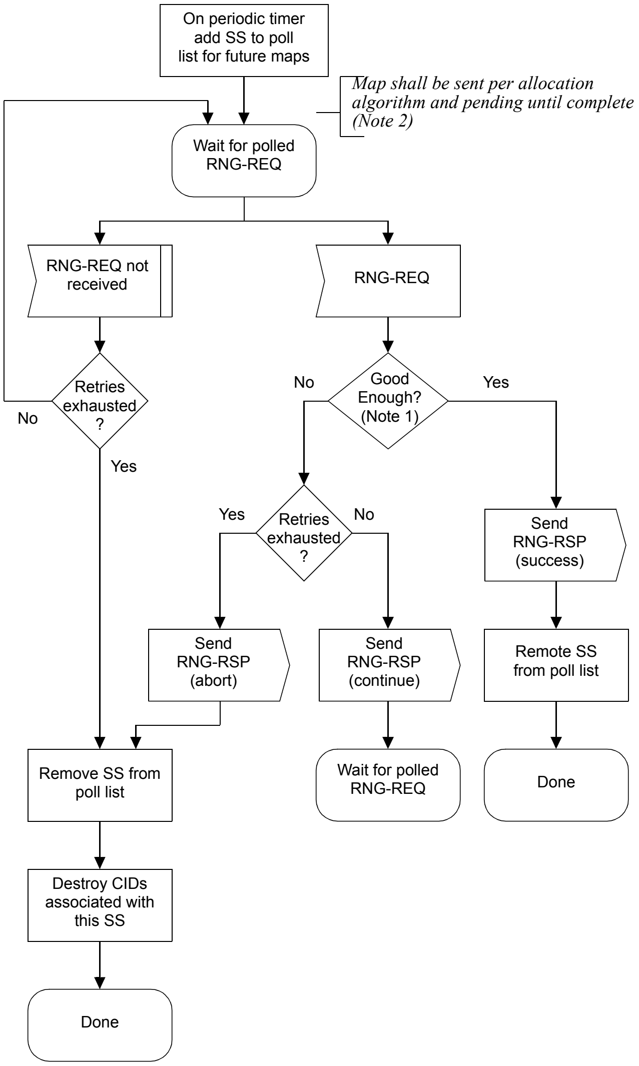
The SS shall reinitialize its MAC sublayer after T4 seconds have elapsed without receiving a Periodic Ranging opportunity. The SS shall also reinitialize its MAC sublayer if it receives a RNG-RSP message from the BS with the Ranging Status field set to 4 (rerange). In both cases, the SS shall also re-register.

Remote RF signal level adjustment at the SS is performed through a station maintenance function using the RNG-REQ and RNG-RSP MAC messages. This is similar to initial ranging and is shown in Figure 55 and Figure 56.

6.2.10.1 Downlink burst profile management in framed operation

The downlink burst profile is determined by the BS according to the quality of the signal that is received by each SS. To reduce the volume of uplink traffic, the SS monitors the carrier to noise and interference ratio [C/(N+I)] and compares the average value against the allowed range of operation. This region is bounded by threshold levels. If the received C/(N+I) goes outside of the allowed operating region, the SS requests a change to a new burst profile using one of three methods. If the SS has a station maintenance interval available, it shall send a RNG-REQ message to which the BS responds with a RNG-RSP message. Otherwise, the SS shall send a DBPC-REQ message in an uplink allocation addressed to that SS's basic connection (regardless of whether the SS is GPC or GPT). The BS responds with a DBPC-RSP message. If neither of these options is available and the SS requires a more robust burst profile on the downlink, the SS shall send a RNG-REQ message in an Initial Maintenance interval. In all three methods, the message is sent using the Basic CID of the SS. The coordination of message transmit and receipt relative to actual change of modulation is different depending upon whether an SS is transitioning to a more or less robust burst profile. Figure 57 shows the case where an SS is transitioning to a more robust type. Figure 58 shows transition to a less robust burst profile.

The SS applies an algorithm to determine its optimal burst profile in accordance with the threshold parameters established in the DCD message in accordance with Figure 59.



NOTES:

1—Means ranging is within the tolerable limits of the BS.

2—RNG-REQ pending-until-complete was nonzero, the BS should hold off the station maintenance opportunity accordingly unless needed, for example, to adjust the SS's power level.

Figure 55—Periodic Ranging—BS

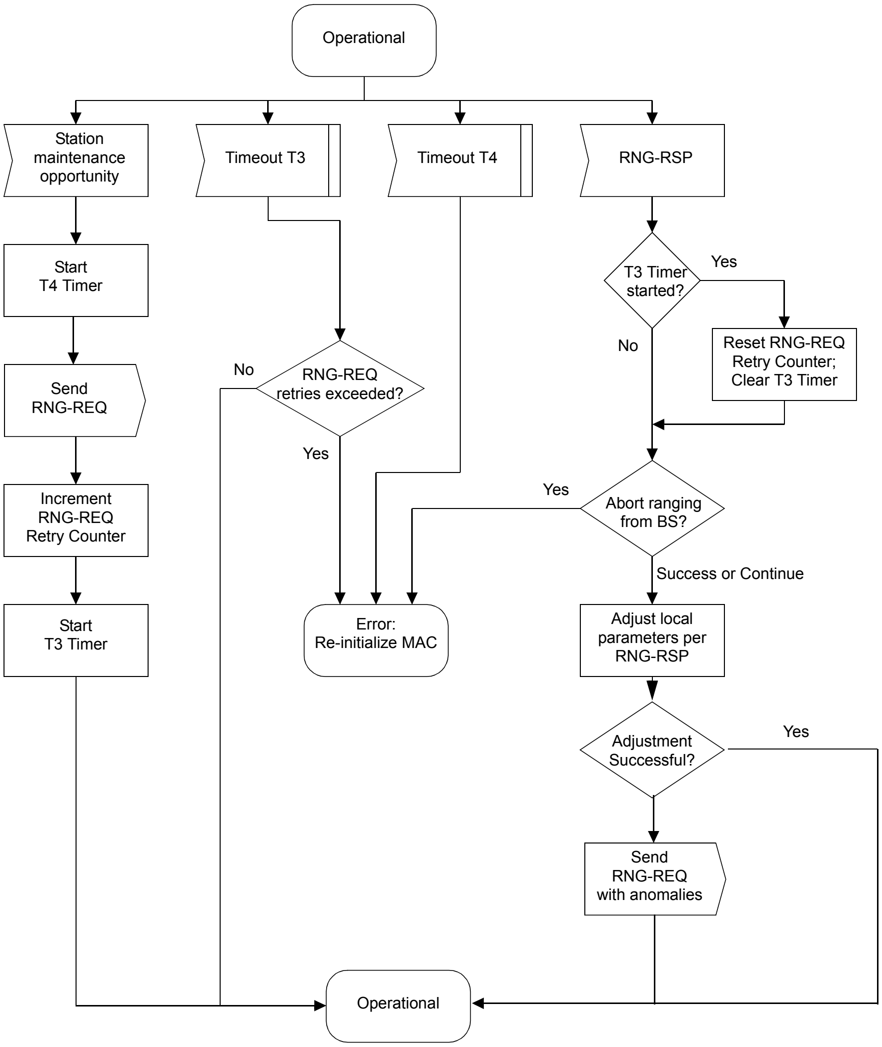


Figure 56—Periodic Ranging—SS

.

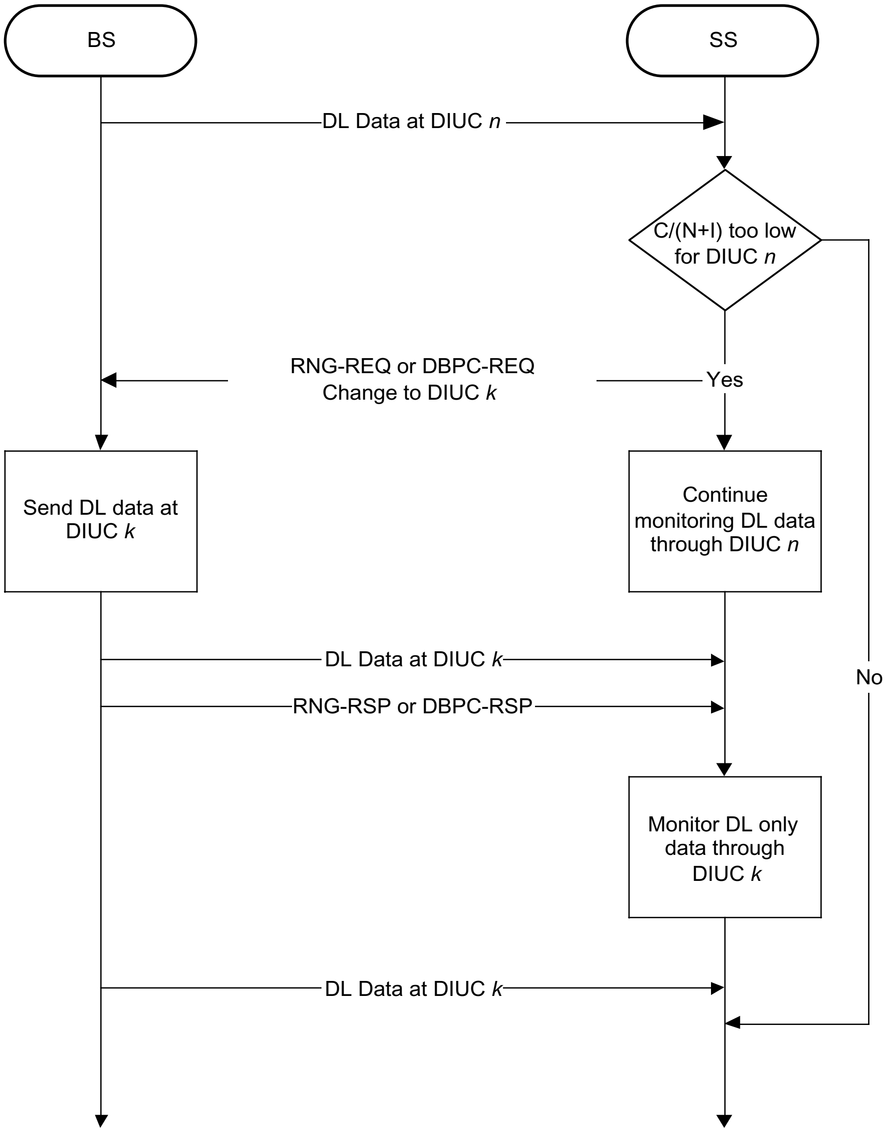


Figure 57—Transition to a more robust burst profile

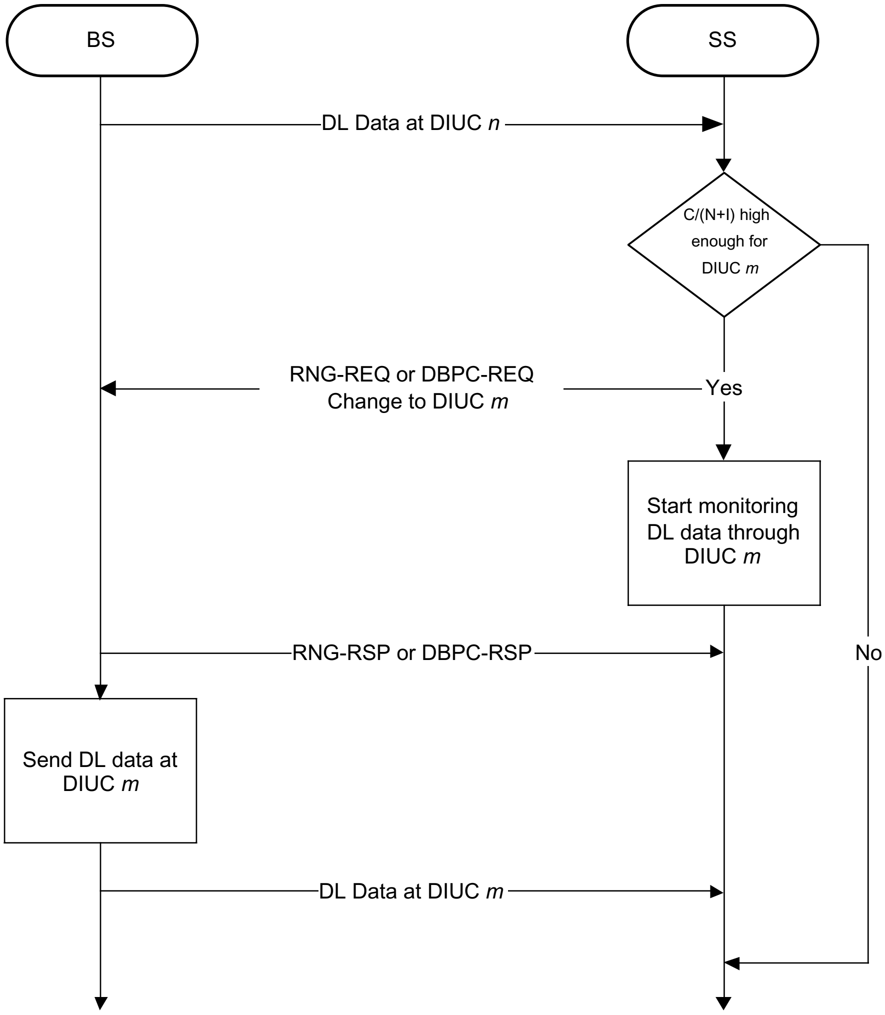

BS
SS
DL Data at DIUC *n*
C/(N+I) high enough for DIUC *m*
RNG-REQ or DBPC-REQ Change to DIUC *m*
Yes
Start monitoring DL data through DIUC *m*
No
RNG-RSP or DBPC-RSP
Send DL data at DIUC *m*
DL Data at DIUC *m*

Figure 58—Transition to a less robust burst profile

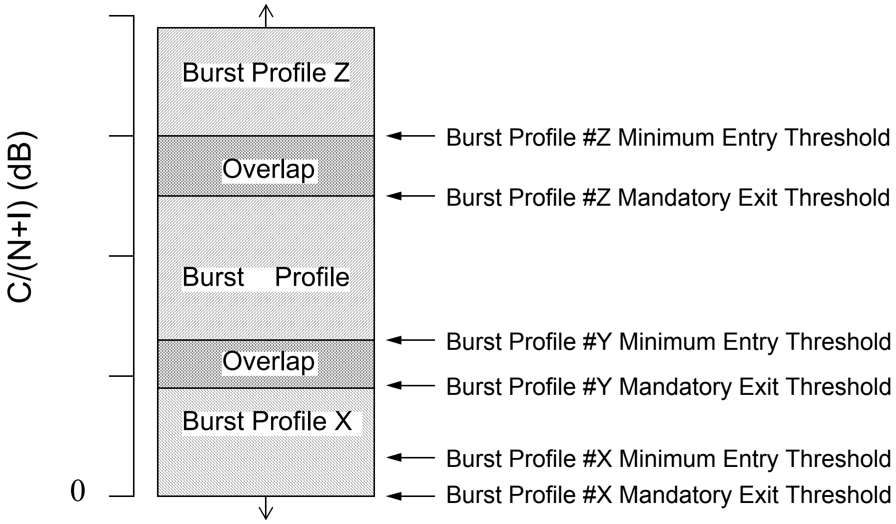

C/(N+I) (dB)
0
Burst Profile Z
Overlap
Burst Profile
Overlap
Burst Profile X
Burst Profile #Z Minimum Entry Threshold
Burst Profile #Z Mandatory Exit Threshold
Burst Profile #Y Minimum Entry Threshold
Burst Profile #Y Mandatory Exit Threshold
Burst Profile #X Minimum Entry Threshold
Burst Profile #X Mandatory Exit Threshold

Figure 59—Burst profile threshold usage

6.2.11 Update of channel descriptors

The channel descriptors (i.e., the UCD and DCD messages) are transmitted at regular intervals by the BS. Each descriptor contains the Configuration Change Count, which shall remain unchanged as long as the channel descriptor remains unchanged. All UL-MAP and DL-MAP messages allocating transmissions and receptions using burst profiles defined in a channel descriptor with a given Configuration Change Count value shall have a UCD/DCD Count value equal to the Configuration Change Count of the corresponding channel descriptor.

The procedure to transition from one generation of the channel descriptors (and, as a consequence, the set of burst profiles) to the next is shown in Table 64 and Table 66, for the uplink and downlink respectively. The Configuration Change Count shall be incremented by 1 modulo 256 for every new generation of channel descriptor. After issuing a DL-MAP or UL-MAP message with the Configuration Change Count equal to that of the new generation, the old channel descriptor ceases to exist and the BS shall not issue UL-MAP and DL-MAP messages referring to it. When transitioning from one generation to the next, the BS shall schedule the transmissions of the UCD and DCD messages in such a way that each terminal has the possibility to hear it at least once.

Table 64—UCD update

BS		SS
send UL-MAP with UCD Count = *i*	-----------UL-MAP------------>	descriptor with UCD Count = *i* previously stored in SS
	<-------------data-----------------	Transmit using burst profiles defined in UCD with Configuration Change Count = *i*
[change of channel descriptor commanded|		
send UL-MAP with UCD Count = *i*	-----------UL-MAP------------>	descriptor with Configuration Change Count = *i* still stored in SS
send UCD message with Configuration Change Count = (*i*+*1* MOD 256)	-----------UCD------------>	store new descriptor with Configuration Change Count = (*i*+*1* MOD 256)
	<--------------data-------------------	Transmit using burst profiles defined in UCD with Configuration Change Count = *i*
send UL-MAP with UCD Count = *i*	-----------UL-MAP------------>	descriptor with Configuration Change Count = *i* still stored in SS
Retransmit UCD message with Configuration Change Count = (*i*+*1* MOD 256) [UCD transition interval start]	-----------UCD------------>	store new descriptor with Configuration Change Count = (*i*+*1* MOD 256)
	<--------------data-------------------	Transmit using burst profiles defined in UCD with Configuration Change Count = *i*
send UL-MAP with UCD Count = *i*	-----------UL-MAP------------>	descriptor with UCD Count = *i* previously stored in SS
	<-------------data-----------------	Transmit using burst profiles defined in UCD with Configuration Change Count = *i*
[UCD transition interval expired]		
send UL-MAP with UCD Count = (*i*+*1* MOD 256)	-----------UL-MAP------------>	delete descriptor with Configuration Change Count = *i*
	<-------------data----------------	Transmit using burst profiles defined in UCD with Configuration Change Count = (*i*+*1* MOD 256)

Table 65—DCD update

BS		SS
send DL-MAP with DCD Count = *i*	-----------DL-MAP------------>	descriptor with Configuration Change Count = *i* previously stored in SS
Transmit using burst profiles defined in DCD with Configuration Change Count = *i*	-------------data----------------->	Receive using burst profiles defined in DCD with Configuration Change Count = *i*
[change of channel descriptor commanded]		
send DL-MAP with DCD Count = *i*	-----------DL-MAP------------>	descriptor with Configuration Change Count = *i* still stored in SS
send DCD message with Configuration Change Count = *(i+1* MOD 256)	-----------DCD------------>	store new descriptor with Configuration Change Count = *(i+1* MOD 256)
Transmit using burst profiles defined in DCD with Configuration Change Count = *i*	-------------data---------------->	Receive using burst profiles defined in DCD with Configuration Change Count = *i*
send DL-MAP with DCD Count = *i*	-----------DL-MAP------------>	descriptor with Configuration Change Count = *i* still stored in SS
Retransmit DCD message with Configuration Change Count = *(i+1* MOD 256) [DCD transition interval start]	-----------DCD------------>	store new descriptor with Configuration Change Count = *(i+1* MOD 256)
Transmit using burst profiles defined in DCD with Configuration Change Count = *i*	-------------data---------------->	Receive using burst profiles defined in DCD with Configuration Change Count = *i*
[DCD transition interval expired]		
send DL-MAP with Configuration Change Count = *(i+1* MOD 256)	-----------DL-MAP------------>	delete descriptor with Configuration Change Count = *i*
Transmit using burst profiles defined in DCD with Configuration Change Count = *i+1*	-------------data---------------->	Receive using burst profiles defined in DCD with Configuration Change Count = *(i+1* MOD 256)

6.2.12 Assigning SSs to multicast groups

The BS may add an SS to a Multicast polling group by sending an MCA-REQ message with the Join command. Upon receiving an MCA-REQ message, the SS shall respond by sending an MCA-RSP message. The protocol is shown in Figure 60 and Figure 61.

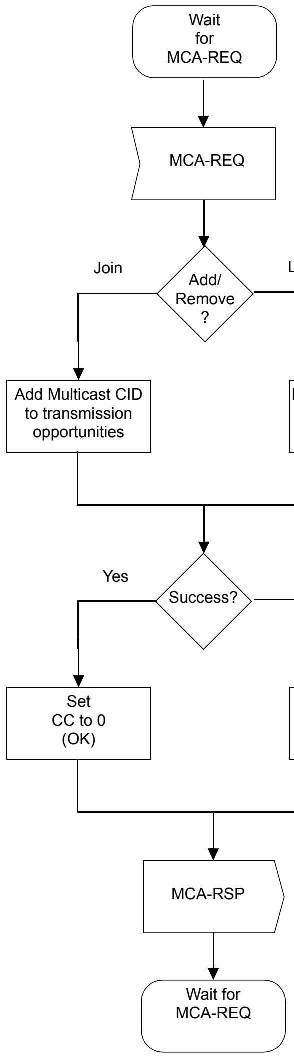


Figure 60—Multicast polling assignment—SS

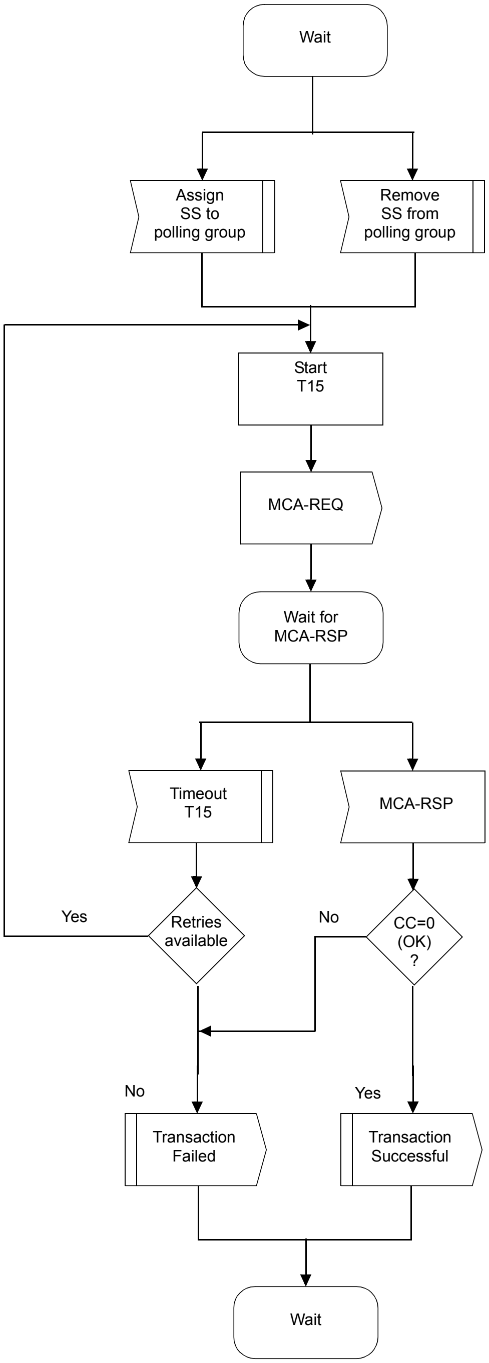


Figure 61—Multicast polling assignment—BS

6.2.13 Quality of Service

This standard defines several QoS related concepts. These include the following:

a) Service Flow QoS Scheduling

b) Dynamic Service Establishment

c) Two-phase Activation Model

6.2.13.1 Theory of operation

The various protocol mechanisms described in this document may be used to support QoS for both uplink and downlink traffic through the SS and the BS. This subclause provides an overview of the QoS protocol mechanisms and their part in providing end-to-end QoS.

The requirements for QoS include the following:

a) A configuration and registration function for preconfiguring SS-based QoS *service flows* and traffic parameters.

b) A signaling function for dynamically establishing QoS-enabled service flows and traffic parameters.

c) Utilization of MAC scheduling and QoS traffic parameters for uplink service flows.

d) Utilization of QoS traffic parameters for downlink service flows.

e) Grouping of service flow properties into named *Service Classes,* so upper-layer entities and external applications (at both the SS and BS) may request service flows with desired QoS parameters in a globally consistent way.

The principal mechanism for providing QoS is to associate packets traversing the MAC interface into a *service flow* as identified by the *CID.* A service flow is a unidirectional flow of packets that is provided a particular QoS. The SS and BS provide this QoS according to the *QoS Parameter Set* defined for the service flow.

The primary purpose of the Quality of Service features defined here is to define transmission ordering and scheduling on the air interface. However, these features often need to work in conjunction with mechanisms beyond the air interface in order to provide end-to-end QoS or to police the behavior of SSs.

Service flows exist in both the uplink and downlink direction and may exist without actually being activated to carry traffic. All service flows have a 32-bit Service Flow Identifier (SFID); active service flows also have a 16-bit CID.

6.2.13.2 Service flows

A *service flow* is a MAC-layer transport service that provides unidirectional transport of packets either to uplink packets transmitted by the SS or to downlink packets transmitted by the BS.[12] A service flow is characterized by a set of *QoS Parameters* such as latency, jitter, and throughput assurances. In order to standardize operation between the SS and BS, these attributes include details of how the SS requests uplink mini-slots and the expected behavior of the BS uplink scheduler.

A service flow is partially characterized by the following attributes:[13]

[12] A service flow, as defined here, has no direct relationship to the concept of a “flow” as defined by the IETF Integrated Services (intserv) Working Group [IETF RFC 2212]. An intserv flow is a collection of packets sharing transport-layer endpoints. Multiple intserv flows can be served by a single service flow.

a) *Service Flow ID:* An SFID is assigned to all existing service flows. The SFID serves as the principal identifier in the SS and the BS for the service flow. A service flow which exists has at least an SFID and an associated Direction.

b) *Connection ID:* Mapping to an SFID exists only when the connection has an admitted service flow(s).

c) *ProvisionedQoSParamSet:* A QoS parameter set provisioned via means outside of the scope of this standard, such as the network management system.

d) *AdmittedQoSParamSet:* Defines a set of QoS parameters for which the BS (and possibly the SS) are reserving resources. The principal resource to be reserved is bandwidth, but this also includes any other memory or time-based resource required to subsequently activate the flow.

e) *ActiveQoSParamSet:* Defines set of QoS parameters defining the service actually being provided to the service flow. Only an Active service flow may forward packets.

f) *Authorization Module:* A logical function within the BS that approves or denies every change to QoS Parameters and Classifiers associated with a service flow. As such it defines an "envelope" that limits the possible values of the AdmittedQoSParamSet and ActiveQoSParamSet.

The relationship between the QoS Parameter Sets is as shown in Figure 62 and Figure 63. The ActiveQoSParamSet is always a subset[14] of the AdmittedQoSParamSet, which is always a subset of the authorized "envelope." In the dynamic authorization model, this envelope is determined by the Authorization Module (labeled as the AuthorizedQoSParamSet). In the provisioned authorization model, this envelope is determined by the ProvisionedQoSParamSet.

It is useful to think of three types of service flows:

a) *Provisioned:* This type of service flow is known via provisioning by, for example, the network management system. Its AdmittedQoSParamSet and ActiveQoSParamSet are both null.

b) *Admitted:* This type of service flow has resources reserved by the BS for its AdmittedQoSParamSet, but these parameters are not active (its ActiveQoSParamSet is null). *Admitted Service Flows* may have been provisioned or may have been signalled by some other mechanism.

c) *Active:* This type of service flow has resources committed by the BS for its ActiveQoSParamSet, (e.g., is actively sending maps containing unsolicited grants for a UGS-based service flow). Its ActiveQoSParamSet is non-null.

[13] Some attributes are derived from the above attribute list. The Service Class Name is an attribute of the ProvisionedQoSParamSet. The activation state of the service flow is determined by the ActiveQoSParamSet. If the ActiveQoSParamSet is null, then the service flow is inactive.

[14] To say that QoS Parameter Set A is a subset of QoS Parameter Set B the following shall be true for all QoS Parameters in A and B:

```
if (a smaller QoS parameter value indicates less resources, e.g., Maximum Traffic Rate)
A is a subset of B if the parameter in A less than or equal to the same parameter in B
if (a larger QoS parameter value indicates less resources, e.g., Tolerated Grant Jitter)
A is a subset of B if the parameter in A is greater than or equal to the same parameter in B
if (the QoS parameter specifies a periodic interval, e.g., Nominal Grant Interval)
A is a subset of B if the parameter in A is an integer multiple of the same parameter in B
if (the QoS parameter is not quantitative, e.g., Service Flow Scheduling Type)
A is a subset of B if the parameter in A is equal to the same parameter in B
```

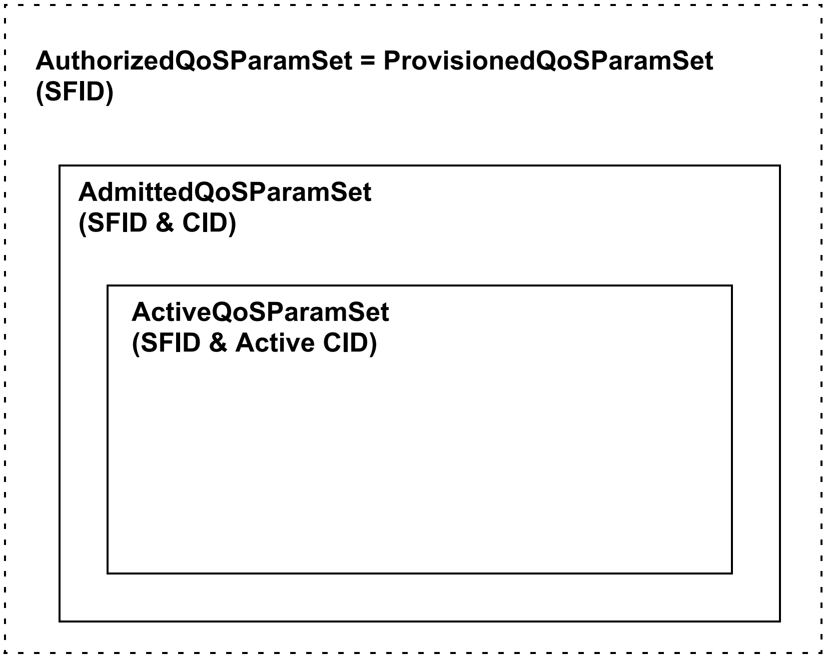


Figure 62—Provisioned authorization model “envelopes”

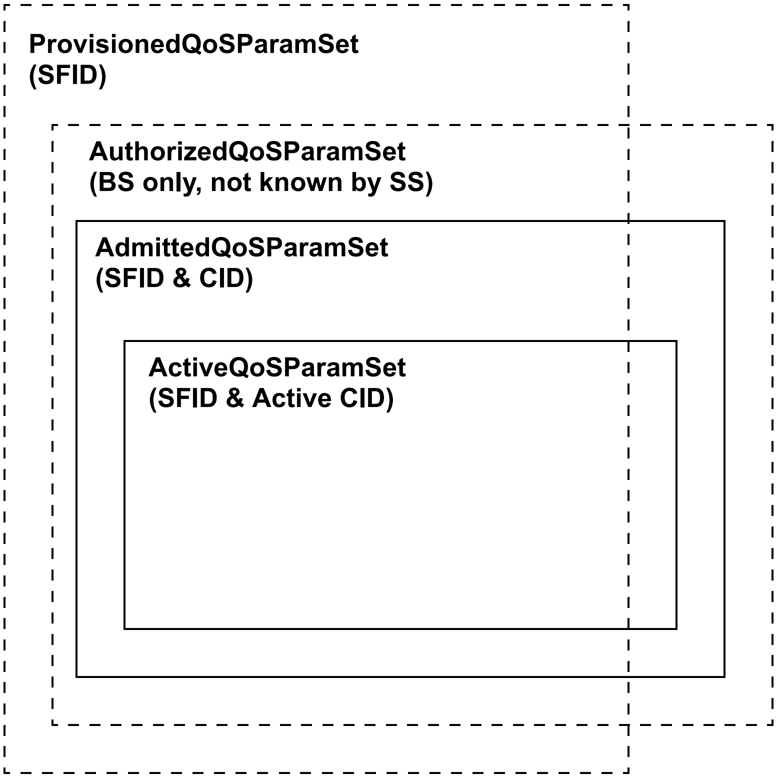


Figure 63—Dynamic authorization model “envelopes”

6.2.13.3 Object model

The major objects of the architecture are represented by named rectangles in Figure 64. Each object has a number of attributes; the attribute names which uniquely identify it are underlined. Optional attributes are denoted with brackets. The relationship between the number of objects is marked at each end of the association line between the objects. For example, a service flow may be associated with from 0 to *N* (many) PDUs, but a PDU is associated with exactly one Service flow. The service flow is the central concept of the MAC protocol. It is uniquely identified by a 32-bit (SFID). Service flows may be in either the uplink or downlink direction. Admitted service flows are mapped a 16-bit CID.

Outgoing user data is submitted to the MAC SAP by a CS process for transmission on the MAC interface. The information delivered to the MAC SAP includes the CID identifying the connection across which the information is delivered. The service flow for the connection is mapped to the CID.

The Service Class is an optional object that may be implemented at the BS. It is referenced by an ASCII name which is intended for provisioning purposes. A Service Class is defined in the BS to have a particular QoS Parameter Set. The QoS Parameter Sets of a service flow may contain a reference to the Service Class Name as a “macro” that selects all of the QoS parameters of the Service Class. The service flow QoS Parameter Sets may augment and even override the QoS parameter settings of the Service Class, subject to authorization by the BS.

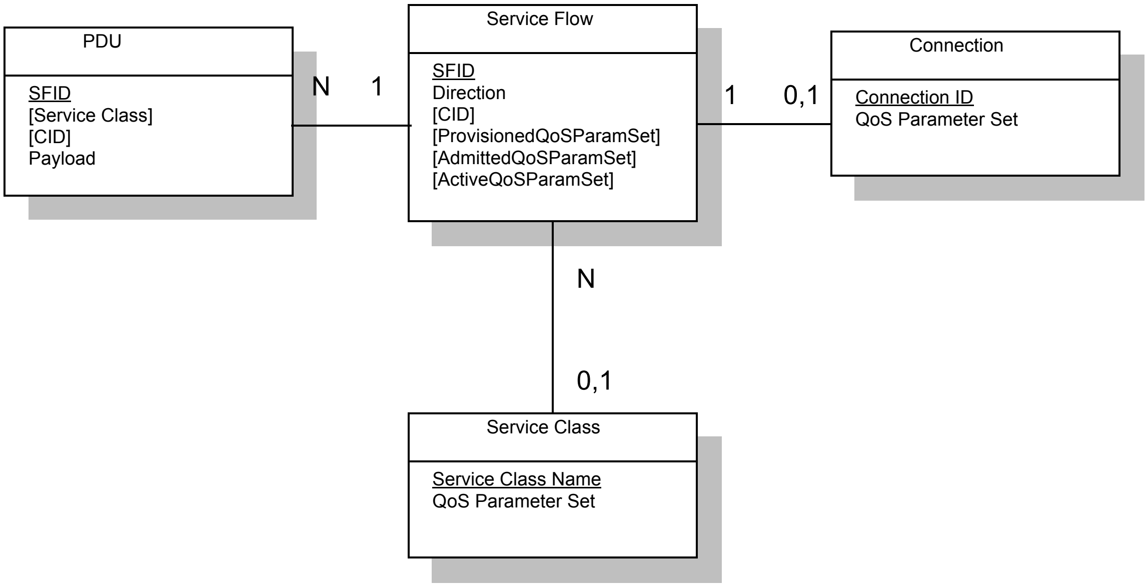


Figure 64—Theory of Operation Object Model

6.2.13.4 Service classes

The Service Class serves the following purposes:

a) It allows operators, who so wish, to move the burden of configuring service flows from the provisioning server to the BS. Operators provision the SSs with the Service Class Name; the implementation of the name is configured at the BS. This allows operators to modify the implementation of a given service to local circumstances without changing SS provisioning. For example, some scheduling parameters may need to be tweaked differently for two different BSs to provide the same service. As another example, service profiles could be changed by time of day.

b) It allows higher-layer protocols to create a service flow by its Service Class Name. For example, telephony signaling may direct the SS to instantiate any available Provisioned service flow of class "G711."

NOTE—Service classes are merely identifiers for a specific set of QoS parameter set values. Hence, the use of service classes is optional. A service identified by a service class is treated no differently, once established, than a service that has the same QoS parameter set explicitly specified.

Any service flow may have its QoS Parameter Set specified in any of three ways:

— By explicitly including all traffic parameters.

— By indirectly referring to a set of traffic parameters by specifying a Service Class Name.

— By specifying a Service Class Name along with modifying parameters.

The Service Class Name is "expanded" to its defined set of parameters at the time the BS successfully admits the service flow. The Service Class expansion can be contained in the following BS-originated Messages: DSA-REQ, DSC-REQ, DSA-RSP, and DSC-RSP. In all of these cases, the BS shall include a service flow encoding that includes the Service Class Name and the QoS Parameter Set of the Service Class. If an SS-initiated request contained any supplemental or overriding service flow parameters, a successful response shall also include these parameters.

When a Service Class name is given in an admission or activation request, it is possible that the returned QoS Parameter Set may change from activation to activation. This can happen because of administrative changes to the Service Class's QoS Parameter Set at the BS. If the definition of a Service Class Name is changed at the BS (e.g., its associated QoS Parameter Set is modified), it has no effect on the QoS Parameters of existing service flows associated with that Service Class. A BS may initiate DSC transactions to existing service flows which reference the Service Class Name to affect the changed Service Class definition.

When an SS uses the Service Class Name to specify the Admitted QoS Parameter Set, the expanded set of TLV encodings of the service flow shall be returned to the SS in the response Message (DSA-RSP or DSC-RSP). Use of the Service Class Name later in the activation request may fail if the definition of the Service Class Name has changed and the new required resources are not available. Thus, the SS should explicitly request the expanded set of TLVs from the response Message in its later activation request.

6.2.13.5 Authorization

Every change to the service flow QoS Parameters shall be approved by an authorization module. This includes every DSA-REQ Message to create a new service flow and every DSC-REQ Message to change a QoS Parameter Set of an existing service flow. Such changes include requesting an admission control decision (e.g., setting the AdmittedQoSParamSet) and requesting activation of a service flow (e.g., setting the ActiveQoSParamSet). Reduction requests regarding the resources to be admitted or activated are also checked by the authorization module.

In the static authorization model, the authorization module stores the provisioned status of all "deferred" service flows. Admission and activation requests for these provisioned service flows shall be permitted, as long as the Admitted QoS Parameter Set is a subset of the Provisioned QoS Parameter Set, and the Active QoS Parameter Set is a subset of the Admitted QoS Parameter Set. Requests to change the Provisioned QoS Parameter Set shall be refused, as shall requests to create new dynamic service flows. This defines a static system where all possible services are defined in the initial configuration of each SS.

In the dynamic authorization model, the authorization module also communicates through a separate interface to an independent policy server. This policy server may provide to the authorization module advance notice of upcoming admission and activation requests, and it specifies the proper authorization action to be taken on those requests. Admission and activation requests from an SS are then checked by the Authorization Module to ensure that the ActiveQoSParamSet being requested is a subset of the set provided by the policy server. Admission and activation requests from an SS that are signalled in advance by the external policy server are permitted. Admission and activation requests from an SS that are not presignalled by the external policy server may result in a real-time query to the policy server or may be refused.

Prior to initial connection setup, the BS shall retrieve the Provisioned QoS Set for an SS. This is handed to the Authorization Module within the BS. The BS shall be capable of caching the Provisioned QoS Parameter Set and shall be able to use this information to authorize dynamic flows which are a subset of the Provisioned QoS Parameter Set. The BS should implement mechanisms for overriding this automated approval process (such as described in the dynamic authorization model). For example it should:

a) Deny all requests whether or not they have been preprovisioned.

b) Define an internal table with a richer policy mechanism but seeded by the configuration file information.

c) Refer all requests to an external policy server.

6.2.13.6 Types of service flows

It is useful to think about three basic types of service flows. This subclause describes these three types of service flows in more detail. However, it is important to note that there are more than just these three basic types (see 11.4.8.5).

6.2.13.6.1 Provisioned service flows

A service flow may be Provisioned but not immediately activated (sometimes called “deferred”). That is, the description of any such service flow contains an attribute which provisions but defers activation and admission (see 11.4.8.5). The BS assigns a SFID for such a service flow but does not reserve resources. The BS may also require an exchange with a policy module prior to admission.

As a result of external action beyond the scope of this specification, the SS may choose to activate a Provisioned service flow by passing the SFID and the associated QoS Parameter Sets to the BS in the DSC-REQ message. If authorized and resources are available, the BS shall respond by mapping the service flow to a CID.

As a result of external action beyond the scope of this specification, the BS may choose to activate a service flow by passing the SFID as well as the CID and the associated QoS Parameter Sets to the SS in the DSC-REQ message. Such a Provisioned service flow may be activated and deactivated many times (through DSC exchanges). In all cases, the original SFID shall be used when reactivating the service flow.

6.2.13.6.2 Admitted service flows

This protocol supports a two-phase activation model which is often utilized in telephony applications. In the two-phase activation model, the resources for a “call” are first “admitted,” and then once the end-to-end negotiation is completed (e.g., called party’s gateway generates an “off-hook” event), the resources are “activated.” Such a two-phase model serves the purposes a) of conserving network resources until a complete end-to-end connection has been established, b) performing policy checks and admission control on resources as quickly as possible, and in particular, before informing the far end of a connection request, and c) preventing several potential theft-of-service scenarios.

For example, if an upper-layer service were using unsolicited grant service, and the addition of upper-layer flows could be adequately provided by increasing The Maximum Sustained Traffic Rate QoS parameter, then the following might be used. When the first higher-layer flow is pending, the SS issues a DSA-REQ with the admitted Maximum Sustained Traffic Rate parameter equal to that required for one higher-layer flow, and the active Maximum Sustained Traffic Rate parameter equal to zero. Later when the higher-layer flow becomes active, it issues a DSC-REQ with the instance of the active Maximum Sustained Traffic Rate parameter equal to that required for one higher layer flow. Admission control was performed at the time of the reservation, so the later DSC-REQ, having the active parameters within the range of the previous reservation, is guaranteed to succeed. Subsequent higher layer flows would be handled in the same way. If there were three higher layer flows establishing connections, with one flow already active, the service flow would have admitted Maximum Sustained Traffic Rate equal to that required for four higher-layer flows, and active Maximum Sustained Traffic Rate equal to that required for one higher-layer flow.

An activation request of a service flow where the new ActiveQoSParamSet is a subset of the AdmittedQoSParamSet shall be allowed, except in the case of catastrophic failure. An admission request where the AdmittedQoSParamSet is a subset of the previous AdmittedQoSParamSet, so long as the ActiveQoSParamSet remains a subset of the AdmittedQoSParamSet, shall succeed.

A service flow that has resources assigned to its AdmittedQoSParamSet but whose resources are not yet completely activated is in a transient state. It is possible in some applications that a long-term reservation of resources is necessary or desirable. For example, placing a telephone call on hold should allow any resources in use for the call to be temporarily allocated to other purposes, but these resources shall be available for resumption of the call later. The AdmittedQoSParamSet is maintained as “soft state” in the BS; this state shall be maintained without releasing the nonactivated resources. Changes may be signaled with a DSC-REQ message.

6.2.13.6.3 Active service flows

A service flow that has a non-NULL ActiveQoSParamSet is said to be an Active service flow. It is requesting (according to its Request/Transmission Policy, as in 11.4.8.12) and being granted bandwidth for transport of data packets. An admitted service flow may be activated by providing an ActiveQoSParamSet, signaling the resources actually desired at the current time. This completes the second stage of the two-phase activation model (see 6.2.13.6.2).

A service flow may be Provisioned and immediately activated. Alternatively, a service flow may be created dynamically and immediately activated. In this case, two-phase activation is skipped and the service flow is available for immediate use upon authorization.

6.2.13.7 Preprovisioned service flow creation

6.2.13.7.1 Static operation

The provisioning of service flows is done via means outside of the scope of this standard, such as the network management system. Configuration of provisioned service flows follows the Registration process. When this is complete, the BS passes service flow encodings to the SS in multiple DSA-REQ messages. The SS replies with DSA-RSP messages to complete service flow initialization.

Service flow Encodings contain either a full definition of service attributes (omitting defaultable items if desired) or a service class name. A service class name is an ASCII string which is known at the BS and which indirectly specifies a set of QoS Parameters.

6.2.13.7.2 Dynamic service flow creation

Service flows may be created by the Dynamic Service Addition (DSA) process as well as through the Registration process outlined above. The DSA may be initiated by either the SS or the BS and may create one uplink and/or one downlink dynamic service flow(s). A three-way handshake is used to create service flows.

6.2.13.7.2.1 Dynamic service flow creation — SS-initiated

The SS-initiated protocol is illustrated in Figure 65 and described in detail in 6.2.13.8.3.1.

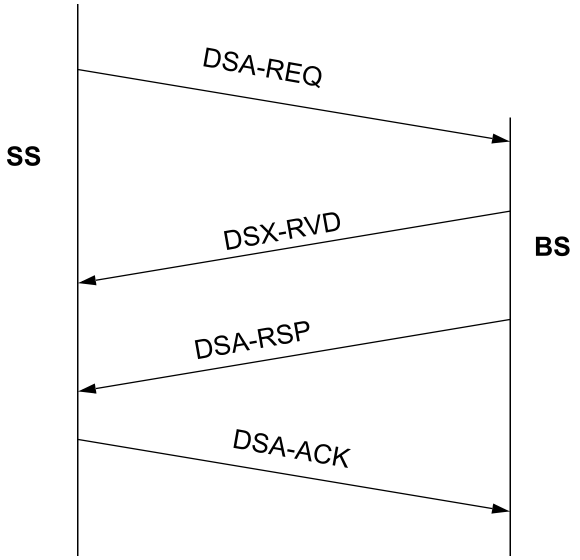


Figure 65—DSA message flow—SS-initiated

A DSA-REQ from an SS contains service flow Reference(s) and QoS Parameter set(s) (marked either for admission-only or for admission and activation).

6.2.13.7.2.2 Dynamic service flow creation—BS-initiated

A DSA-REQ from a BS contains Service Flow Identifier(s) for one uplink and/or one downlink Service flow, possibly their associated CIDs, and set(s) of active or admitted QoS Parameters. The protocol is illustrated in Figure 66 and is described in detail in 6.2.13.8.3.2.

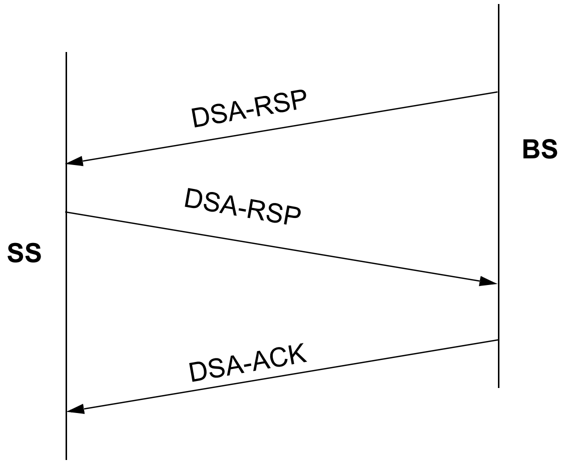


Figure 66—DSA message flow—BS-initiated

6.2.13.7.2.3 Dynamic service flow modification and deletion

In addition to the methods presented above for creating service flows, protocols are defined for modifying and deleting service flows; see 6.2.13.8.4 and 6.2.13.8.5.

Both provisioned and dynamically created Service flows are modified with the DSC message, which can change the Admitted and Active QoS Parameter sets of the flow.

A successful DSC transaction changes a service flow's QoS parameters by replacing both the Admitted and Active QoS parameter sets. If the message contains only the Admitted set, the Active set is set to null and the flow is deactivated. If the message contains neither set ("000" value used for QoS Parameter Set type, see 11.4.8.5), then both sets are set to null and the flow is deadmitted. When the message contains both QoS parameter sets, the Admitted set is checked first, and if admission control succeeds, the Active set in the message is checked against the Admitted set in the message to ensure that it is a subset. If all checks are successful, the QoS parameter sets in the message become the new Admitted and Active QoS parameter sets for the service flow. If either of the checks fails, the DSC transaction fails and the service flow QoS parameter sets are unchanged.

6.2.13.8 Dynamic service

6.2.13.8.1 Connection establishment

Service flows may be created, changed, or deleted. This is accomplished through a series of MAC management Messages referred to as Dynamic Service Addition (DSA), Dynamic Service Change (DSC), and Dynamic Service Deletion (DSD). The DSA Messages create a new service flow. The DSC Messages change an existing service flow. The DSD Messages delete an existing service flow. This is illustrated in Figure 67.

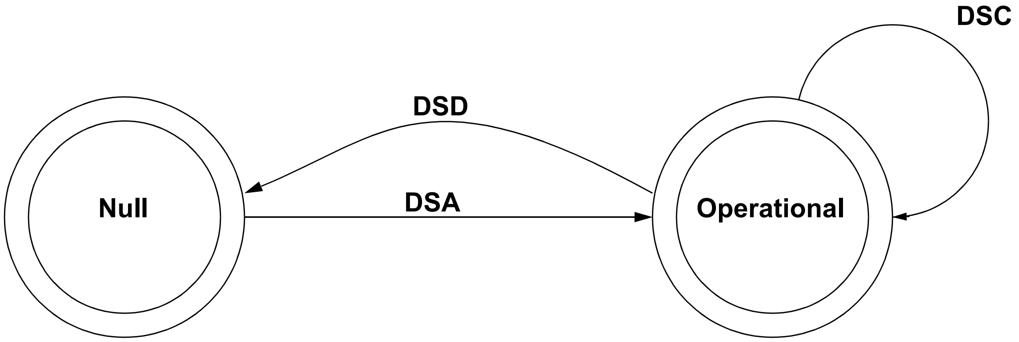


Figure 67—Dynamic service flow overview

The Null state implies that no service flow exists that matches the SFID and/or Transaction ID in a Message. Once the service flow exists, it is operational and has an assigned SFID. In steady state operation, a service flow resides in a Nominal state. When Dynamic Service messaging is occurring, the service flow may transition through other states, but remains operational. Since multiple service flows may exist, there may be multiple state machines active, one for every service flow. Dynamic Service Messages only affect those state machines that match the SFID and/or Transaction ID. If privacy is enabled, both the SS and BS shall verify the HMAC digest on all dynamic service Messages before processing them, and discard any Messages that fail.

Transaction IDs are unique per transaction and are selected by the initiating device (SS or BS). To help prevent ambiguity and provide simple checking, the Transaction ID number space is split between the SS and BS. The SS shall select its Transaction IDs from the first half of the number space (0x0000 to 0x7FFF). The BS shall select its Transaction IDs from the second half of the number space (0x8000 to 0xFFFF).

Each dynamic service Message sequence is a unique transaction with an associated unique transaction identifier. The DSA/DSC transactions consist of a request/response/acknowledge sequence. The DSD transactions consist of a request/response sequence. The response Messages shall return a confirmation code of OK unless some exception condition was detected. The acknowledge Messages shall return the confirmation code in the response unless a new exception condition arises. A more detailed state diagram, including transition states, is shown below. The detailed actions for each transaction shall be given in the following subclauses.

6.2.13.8.2 Dynamic service flow state transitions

The Dynamic Service Flow State Transition Diagram (Figure 68) is the top-level state diagram and controls the general service flow state. As needed, it creates transactions, each represented by a Transaction state transition diagram, to provide the DSA, DSC, and DSD signaling. Each Transaction state transition diagram communicates only with the parent Dynamic Service Flow State Transition Diagram. The top-level state transition diagram filters Dynamic Service Messages and passes them to the appropriate transaction based on SFID, Service Flow Reference number, and Transaction ID.

There are six different types of transactions: locally initiated or remotely initiated for each of the DSA, DSC, and DSD Messages (Figure 69–Figure 74). Most transactions have three basic states: pending, holding, and deleting. The pending state is typically entered after creation and is where the transaction is waiting for a reply. The holding state is typically entered once the reply is received. The purpose of this state is to allow for retransmissions in case of a lost Message, even though the local entity has perceived that the transaction has completed. The deleting state is only entered if the service flow is being deleted while a transaction is being processed.

The flow diagrams provide a detailed representation of each of the states in the Transaction state transition diagrams. All valid transitions are shown. Any inputs not shown should be handled as a severe error condition.

With one exception, these state diagrams apply equally to the BS and SS. In the Dynamic Service Flow Changing-Local state, there is a subtle difference in the SS and BS behaviors. This is called out in the state transition and detailed flow diagrams.

NOTE—The ‘Num Xacts’ variable in the Dynamic Service Flow State Transition Diagram is incremented every time the top-level state diagram creates a transaction and is decremented every time a transaction terminates. A Dynamic Service Flow shall not return to the Null state until it’s deleted and all transactions have terminated.

The inputs for the state diagrams are identified below.

Dynamic Service Flow State Transition Diagram inputs from unspecified local, higher-level entities:

a) Add
b) Change
c) Delete

Dynamic Service Flow State Transition Diagram inputs from DSx Transaction State Transition diagrams:

a) DSA Succeeded
b) DSA Failed
c) DSA ACK Lost
d) DSA Erred
e) DSA Ended

a) DSC Succeeded
b) DSC Failed
c) DSC ACK Lost
d) DSC Erred
e) DSC Ended

a) DSD Succeeded
b) DSD Erred
c) DSD Ended

DSx Transaction State Transition diagram inputs from the Dynamic Service Flow State Transition Diagram:

a) SF Add
b) SF Change
c) SF Delete

a) SF Abort Add
b) SF Change-Remote
c) SF Delete-Local
d) SF Delete-Remote

a) SF DSA-ACK Lost

b) SF DSC-REQ Lost

c) SF DSC-ACK Lost

d) SF DSC-REQ Lost

a) SF Changed

b) SF Deleted

The creation of DSx Transactions by the Dynamic Service Flow State Transition Diagram is indicated by the notation:

DSx-[Local | Remote] (initial_input)

where initial_input may be SF Add, DSA-REQ, SF Change, DSC-REQ, SF Delete, or DSD-REQ, depending on the transaction type and initiator.

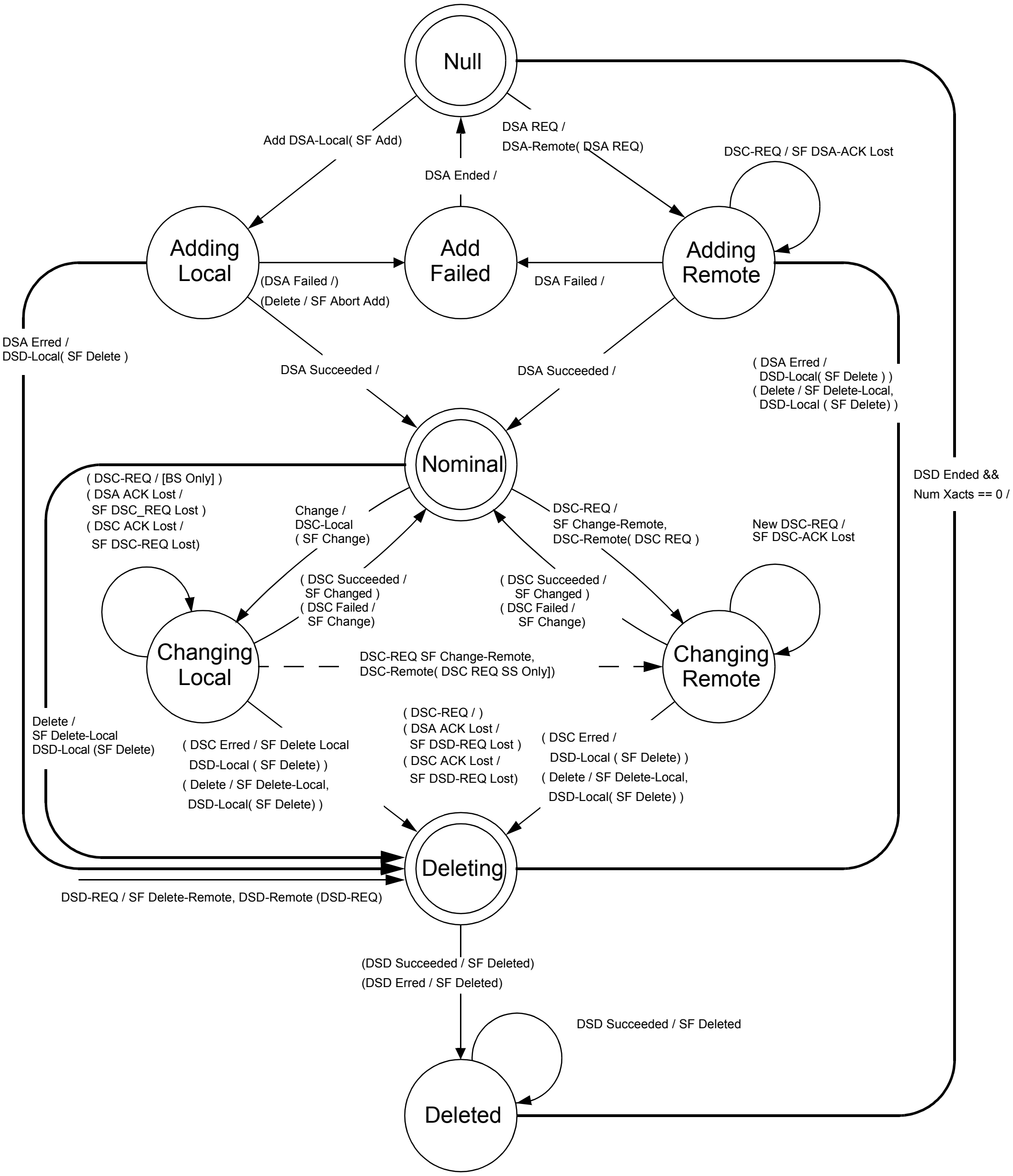


Figure 68—Dynamic service flow state transition diagram

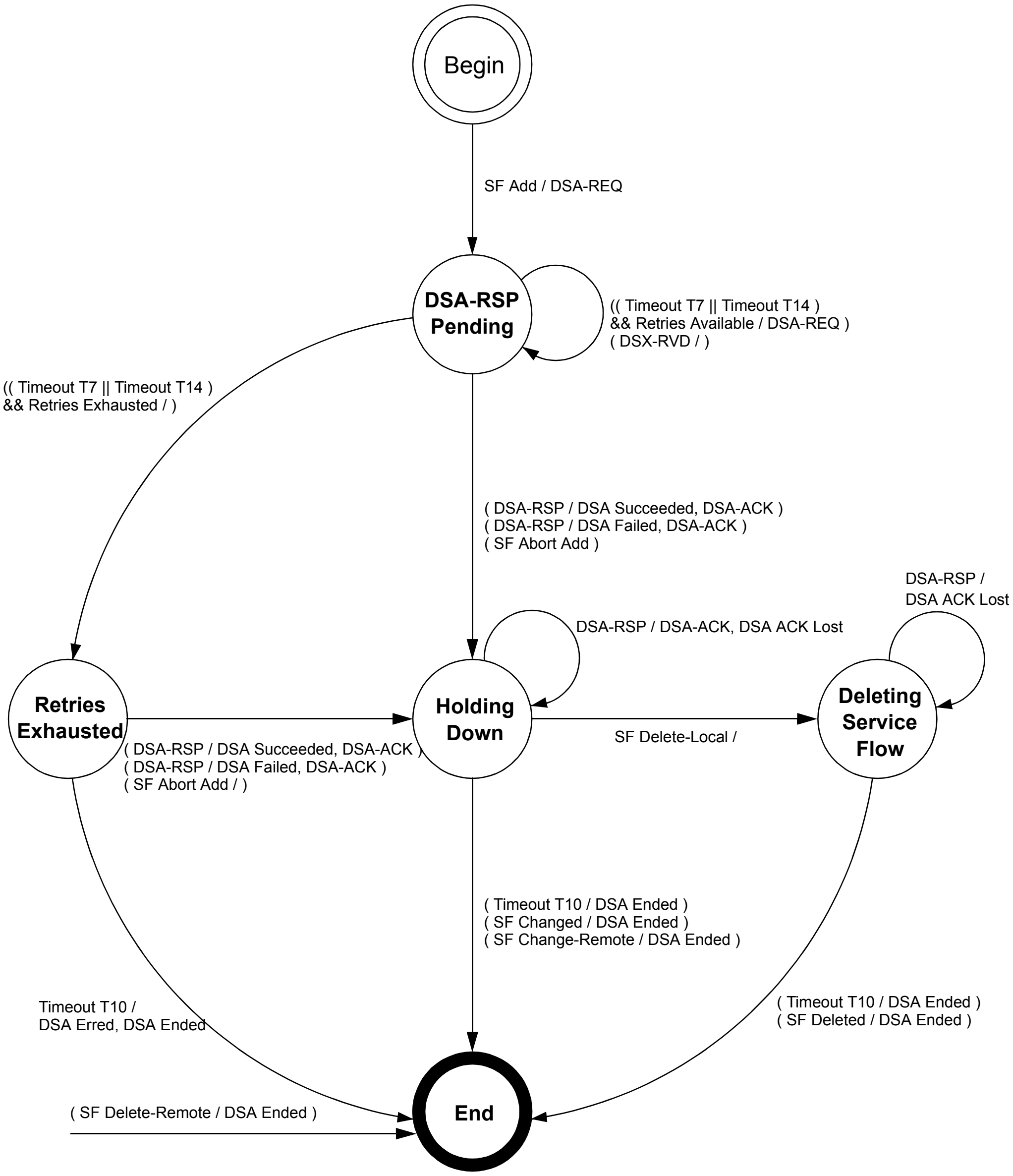

Begin
SF Add / DSA-REQ
DSA-RSP Pending
((Timeout T7 || Timeout T14) && Retries Available / DSA-REQ)
(DSX-RVD /)
((Timeout T7 || Timeout T14) && Retries Exhausted /)
(DSA-RSP / DSA Succeeded, DSA-ACK)
(DSA-RSP / DSA Failed, DSA-ACK)
(SF Abort Add)
DSA-RSP / DSA ACK Lost
DSA-RSP / DSA-ACK, DSA ACK Lost
Retries Exhausted
Holding Down
Deleting Service Flow
SF Delete-Local /
(DSA-RSP / DSA Succeeded, DSA-ACK)
(DSA-RSP / DSA Failed, DSA-ACK)
(SF Abort Add /)
(Timeout T10 / DSA Ended)
(SF Changed / DSA Ended)
(SF Change-Remote / DSA Ended)
Timeout T10 / DSA Erred, DSA Ended
(Timeout T10 / DSA Ended)
(SF Deleted / DSA Ended)
(SF Delete-Remote / DSA Ended)
End

Figure 69—DSA—Locally initiated transaction state transition diagram

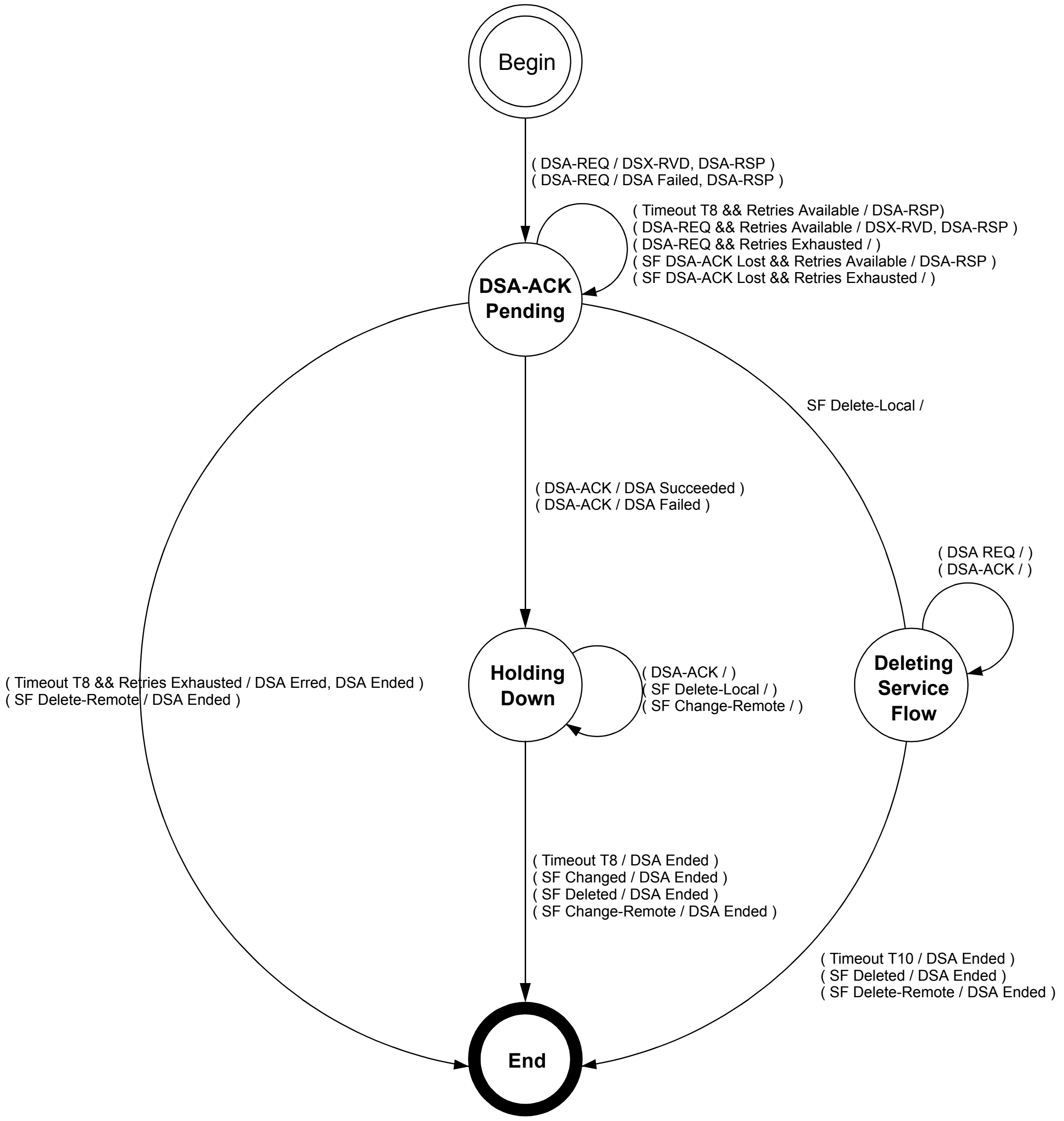

Begin
(DSA-REQ / DSX-RVD, DSA-RSP)
(DSA-REQ / DSA Failed, DSA-RSP)
(Timeout T8 && Retries Available / DSA-RSP)
(DSA-REQ && Retries Available / DSX-RVD, DSA-RSP)
(DSA-REQ && Retries Exhausted /)
(SF DSA-ACK Lost && Retries Available / DSA-RSP)
(SF DSA-ACK Lost && Retries Exhausted /)
DSA-ACK Pending
SF Delete-Local /
(DSA-ACK / DSA Succeeded)
(DSA-ACK / DSA Failed)
(DSA REQ /)
(DSA-ACK /)
(Timeout T8 && Retries Exhausted / DSA Erred, DSA Ended)
(SF Delete-Remote / DSA Ended)
Holding Down
(DSA-ACK /)
(SF Delete-Local /)
(SF Change-Remote /)
Deleting Service Flow
(Timeout T8 / DSA Ended)
(SF Changed / DSA Ended)
(SF Deleted / DSA Ended)
(SF Change-Remote / DSA Ended)
(Timeout T10 / DSA Ended)
(SF Deleted / DSA Ended)
(SF Delete-Remote / DSA Ended)
End

Figure 70—DSA—Remotely initiated transaction state transition diagram

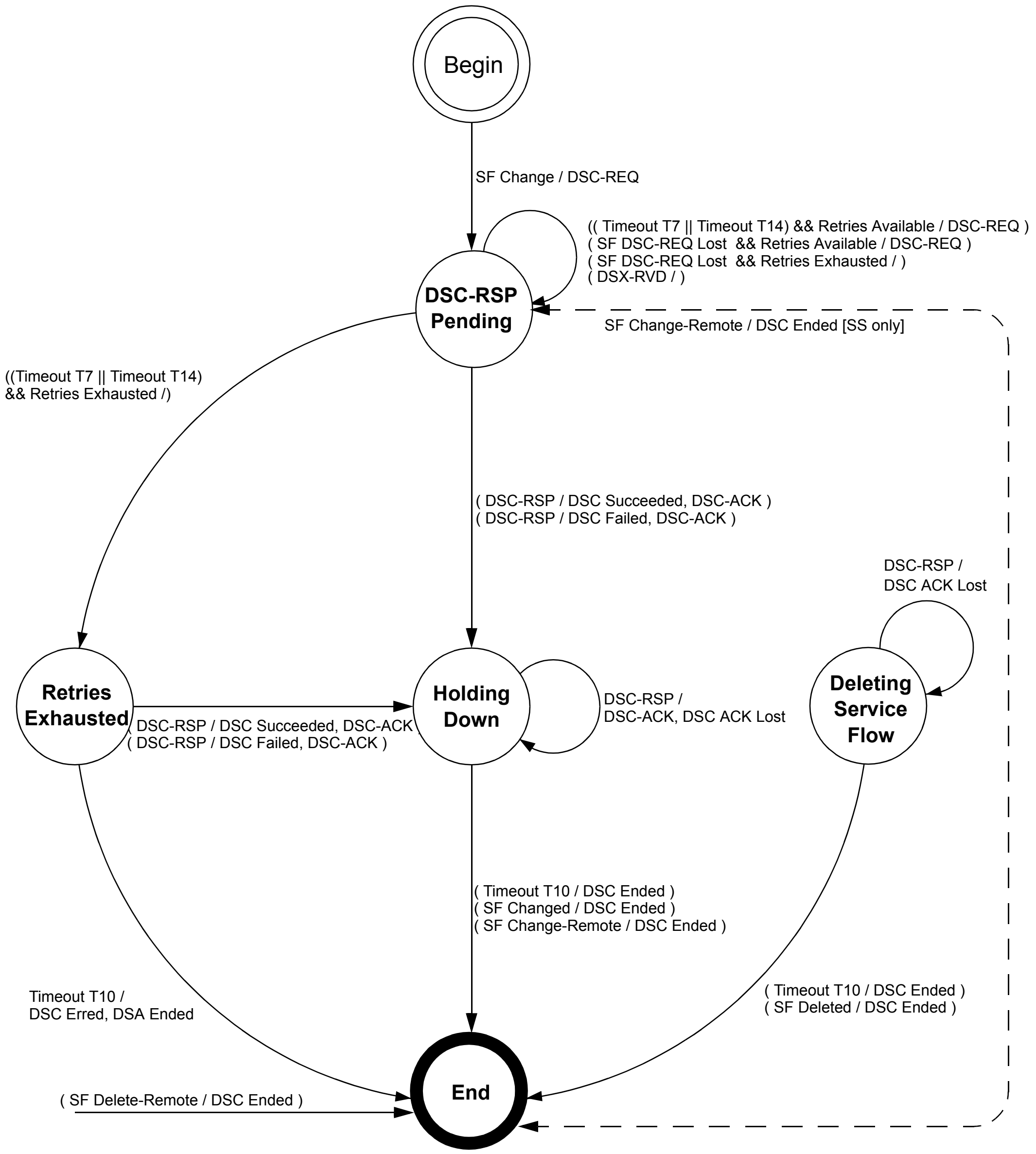

Begin
SF Change / DSC-REQ
((Timeout T7 || Timeout T14) && Retries Available / DSC-REQ)
(SF DSC-REQ Lost && Retries Available / DSC-REQ)
(SF DSC-REQ Lost && Retries Exhausted /)
(DSX-RVD /)
DSC-RSP Pending
SF Change-Remote / DSC Ended [SS only]
((Timeout T7 || Timeout T14)
&& Retries Exhausted /)
(DSC-RSP / DSC Succeeded, DSC-ACK)
(DSC-RSP / DSC Failed, DSC-ACK)
DSC-RSP /
DSC ACK Lost
Retries Exhausted
(DSC-RSP / DSC Succeeded, DSC-ACK)
(DSC-RSP / DSC Failed, DSC-ACK)
Holding Down
DSC-RSP /
DSC-ACK, DSC ACK Lost
Deleting Service Flow
(Timeout T10 / DSC Ended)
(SF Changed / DSC Ended)
(SF Change-Remote / DSC Ended)
Timeout T10 /
DSC Erred, DSA Ended
(Timeout T10 / DSC Ended)
(SF Deleted / DSC Ended)
(SF Delete-Remote / DSC Ended)
End

Figure 71—DSC—Locally initiated transaction state transition diagram

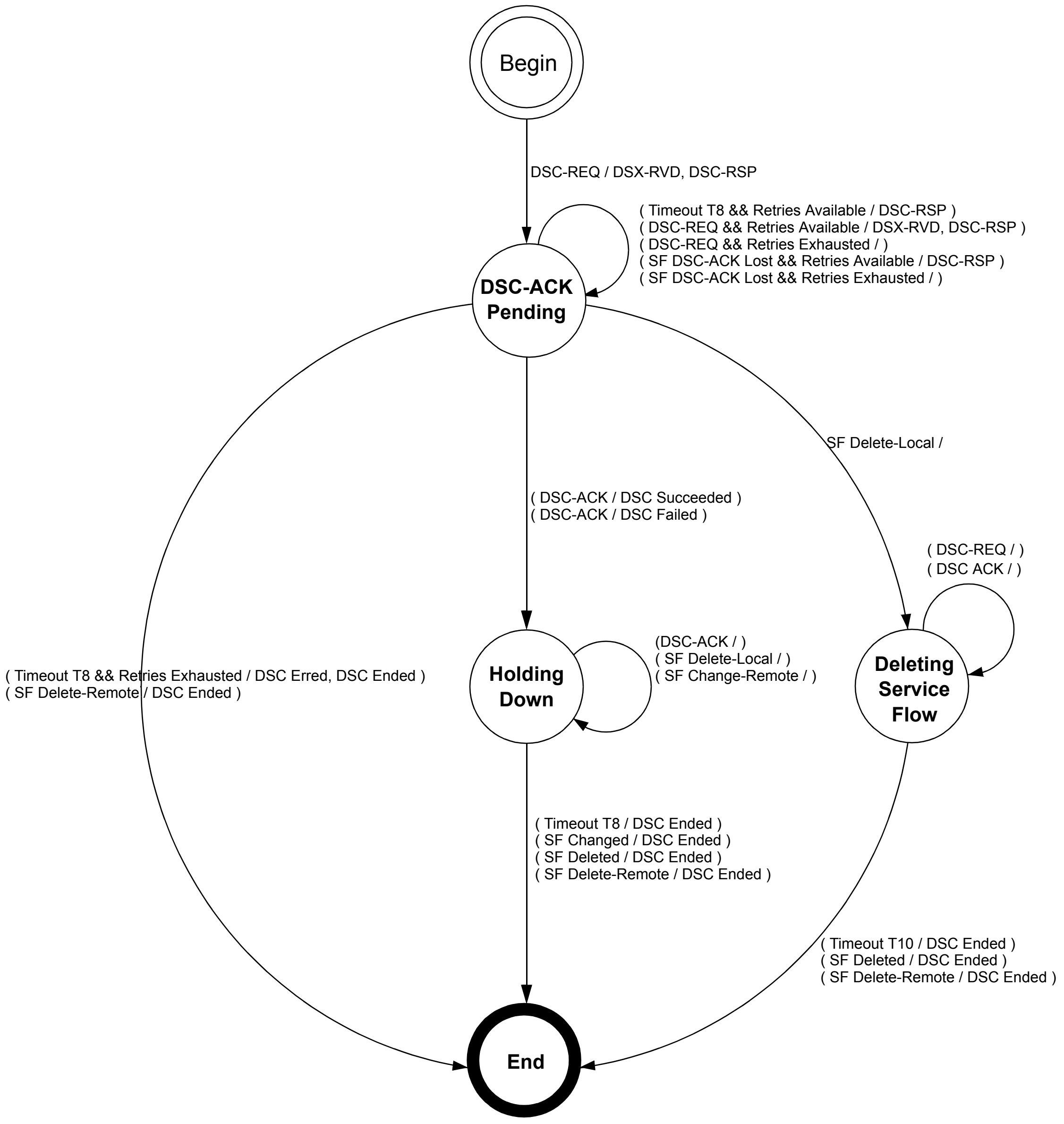

Begin
DSC-REQ / DSX-RVD, DSC-RSP
(Timeout T8 && Retries Available / DSC-RSP)
(DSC-REQ && Retries Available / DSX-RVD, DSC-RSP)
(DSC-REQ && Retries Exhausted /)
(SF DSC-ACK Lost && Retries Available / DSC-RSP)
(SF DSC-ACK Lost && Retries Exhausted /)
DSC-ACK Pending
SF Delete-Local /
(DSC-ACK / DSC Succeeded)
(DSC-ACK / DSC Failed)
(DSC-REQ /)
(DSC ACK /)
(DSC-ACK /)
(SF Delete-Local /)
(SF Change-Remote /)
(Timeout T8 && Retries Exhausted / DSC Erred, DSC Ended)
(SF Delete-Remote / DSC Ended)
Holding Down
Deleting Service Flow
(Timeout T8 / DSC Ended)
(SF Changed / DSC Ended)
(SF Deleted / DSC Ended)
(SF Delete-Remote / DSC Ended)
(Timeout T10 / DSC Ended)
(SF Deleted / DSC Ended)
(SF Delete-Remote / DSC Ended)
End

Figure 72—DSC—Remotely initiated transaction state transition diagram

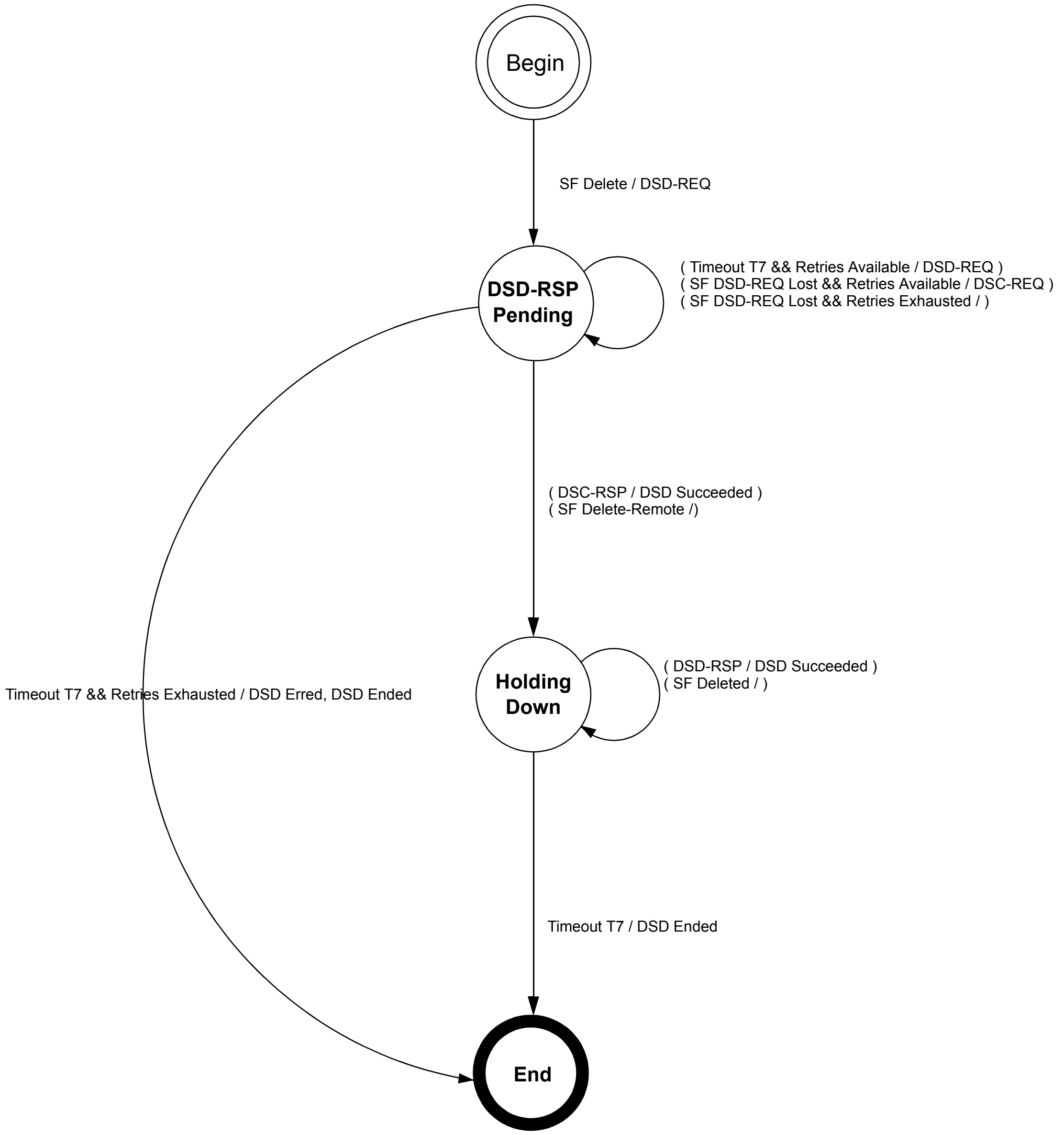

Begin
SF Delete / DSD-REQ
DSD-RSP Pending
(Timeout T7 && Retries Available / DSD-REQ)
(SF DSD-REQ Lost && Retries Available / DSC-REQ)
(SF DSD-REQ Lost && Retries Exhausted /)
(DSC-RSP / DSD Succeeded)
(SF Delete-Remote /)
Timeout T7 && Retries Exhausted / DSD Erred, DSD Ended
Holding Down
(DSD-RSP / DSD Succeeded)
(SF Deleted /)
Timeout T7 / DSD Ended
End

Figure 73—DSD—Locally initiated transaction state transition diagram

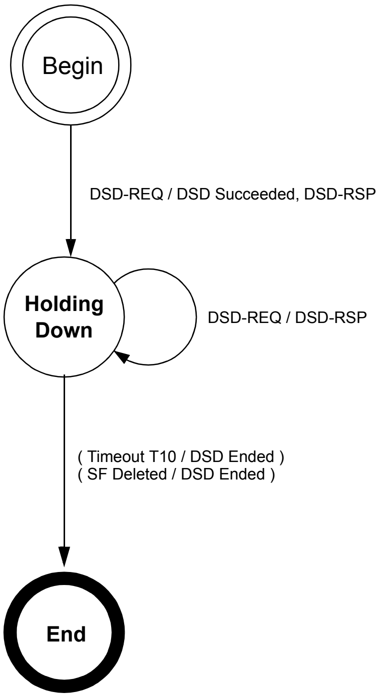

Begin
DSD-REQ / DSD Succeeded, DSD-RSP
Holding Down
DSD-REQ / DSD-RSP
(Timeout T10 / DSD Ended)
(SF Deleted / DSD Ended)
End

Figure 74—DSD—Remotely initiated transaction state transition diagram

6.2.13.8.3 Dynamic Service Addition (DSA)

6.2.13.8.3.1 SS-initiated DSA

An SS wishing to create an uplink and/or downlink Service flow sends a request to the BS using a dynamic service addition request message (DSA-REQ). The BS checks the integrity of the message and, if the message is intact, sends a message received (DSX-RVD) response to the SS. The BS checks the SS's authorization for the requested service(s) and whether the QoS requirements can be supported, generating an appropriate response using a dynamic service addition response Message (DSA-RSP). The SS concludes the transaction with an acknowledgment Message (DSA-ACK). This process is illustrated in Table 66.

In order to facilitate a common admission response, an uplink and a downlink Service flow can be included in a single DSA-REQ. Both service flows are either accepted or rejected together.

Table 66—DSA initiated from SS

SS		BS
New service flow(s) needed		
Check if resources are available		
Send DSA-REQ Set Timers T7 and T14	---DSA-REQ-->	Receive DSA-REQ
Timer T14 Stops	<-- DSX-RVD--	DSA-REQ integrity valid
		Check whether SS is authorized for Service(s)[a]
		Check whether service flow(s) QoS can be supported
		Create SFID(s)
		If uplink AdmittedQoSParamSet is non-null, map service flow to CID
		If uplink ActiveQoSParamSet is non-null, Enable reception of data on new uplink service flow
Receive DSA-RSP Timer T7 Stops	<--DSA-RSP---	Send DSA-RSP
If ActiveQoSParamSet is non-null, Enable transmission and/or reception of data on new service flow(s)		
Send DSA-ACK	---DSA-ACK-->	Receive DSA-ACK
		If downlink ActiveQoSParamSet is non-null, Enable transmission of data on new downlink service flow

[a] Authorization happens prior to the DSA-REQ being received by the BS. The details of BS signalling to anticipate a DSA-REQ are beyond the scope of this specification.

A BS wishing to establish an uplink and/or a downlink dynamic service flow(s) with an SS performs the following operations. The BS checks the authorization of the destination SS for the requested class of service and to determine whether the QoS requirements can be supported. If the service can be supported, the BS generates new SFID(s) with the required class of service and informs the SS using a DSA-REQ Message. If the SS checks that it can support the service, it responds using a DSA Message. The transaction completes with the BS sending the acknowledge Message (DSA-ACK). This process is illustrated in Table 67.

Table 67—DSA initiated from BS

SS		BS
		New service flow(s) required for SS
		Check whether SS is authorized for Service(s)
		Check whether service flow(s) QoS can be supported
		Create SFID(s)
		If AdmittedQoSParamSet is non-null, map service flow to CID
Receive DSA-REQ	<--DSA-REQ---	Send DSA-REQ Set Timer T7
Confirm that SS can support service flow(s)		
Add Downlink SFID (if present)		
Enable reception on any new downlink service flow		
Send DSA-RSP	---DSA-RSP-->	Receive DSA-RSP Timer T7 Stops
		Enable transmission (downlink) or reception (uplink) of data on new service flow
Receive DSA-ACK	<--DSA-ACK---	Send DSA-ACK
Enable transmission on new uplink service flow		

6.2.13.8.3.2 DSA state transition diagrams

DSA state transition diagrams are shown in Figure 75—Figure 83.

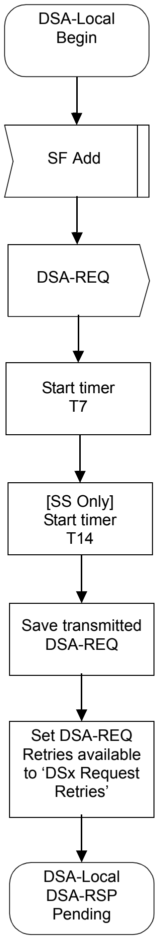

DSA-Local Begin
SF Add
DSA-REQ
Start timer T7
[SS Only] Start timer T14
Save transmitted DSA-REQ
Set DSA-REQ Retries available to ‘DSx Request Retries’
DSA-Local DSA-RSP Pending

Figure 75—DSA—Locally initiated transaction begin state flow diagram

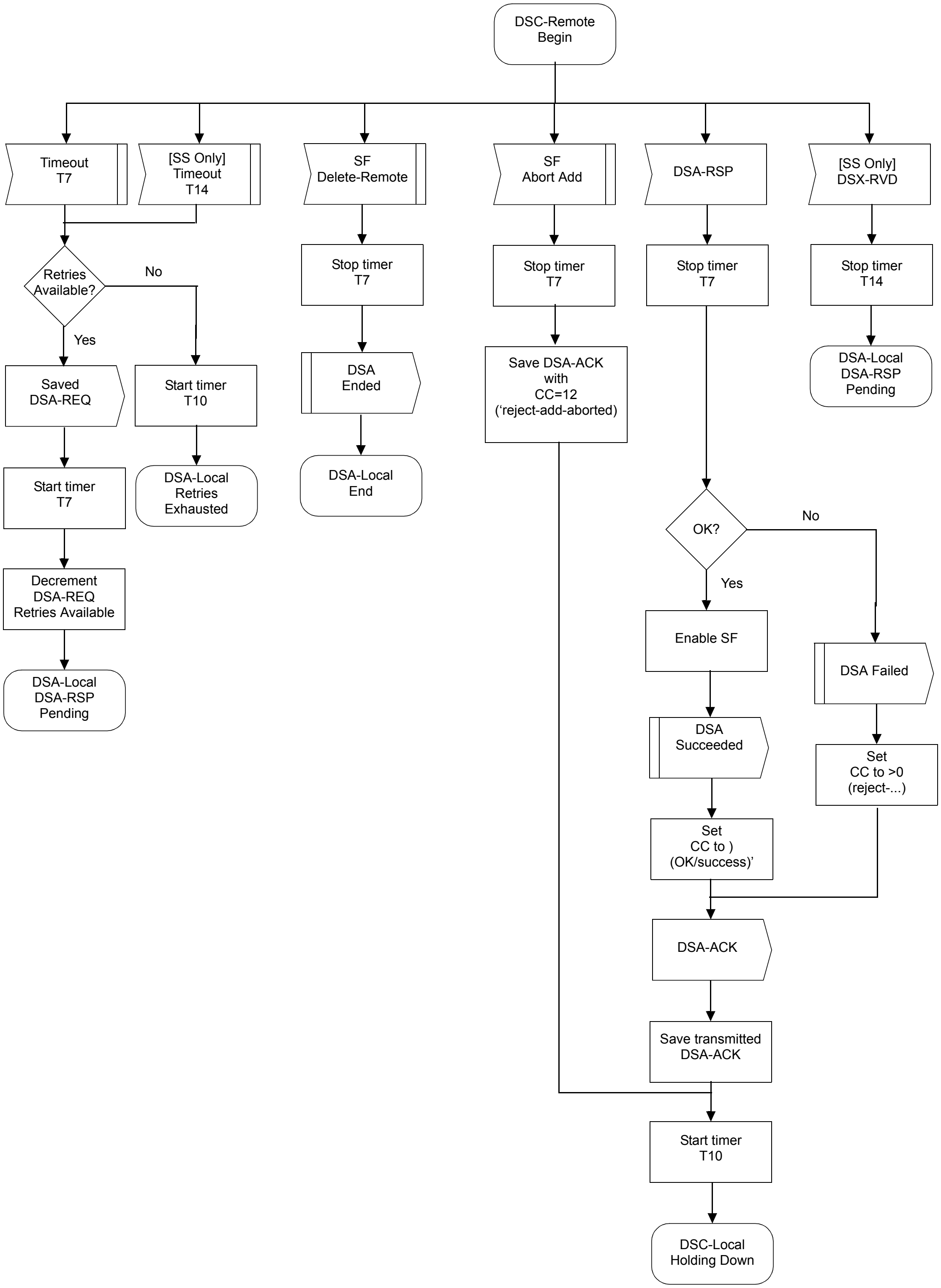


Figure 76—DSA—Locally initiated transaction DSA-RSP pending state flow diagram

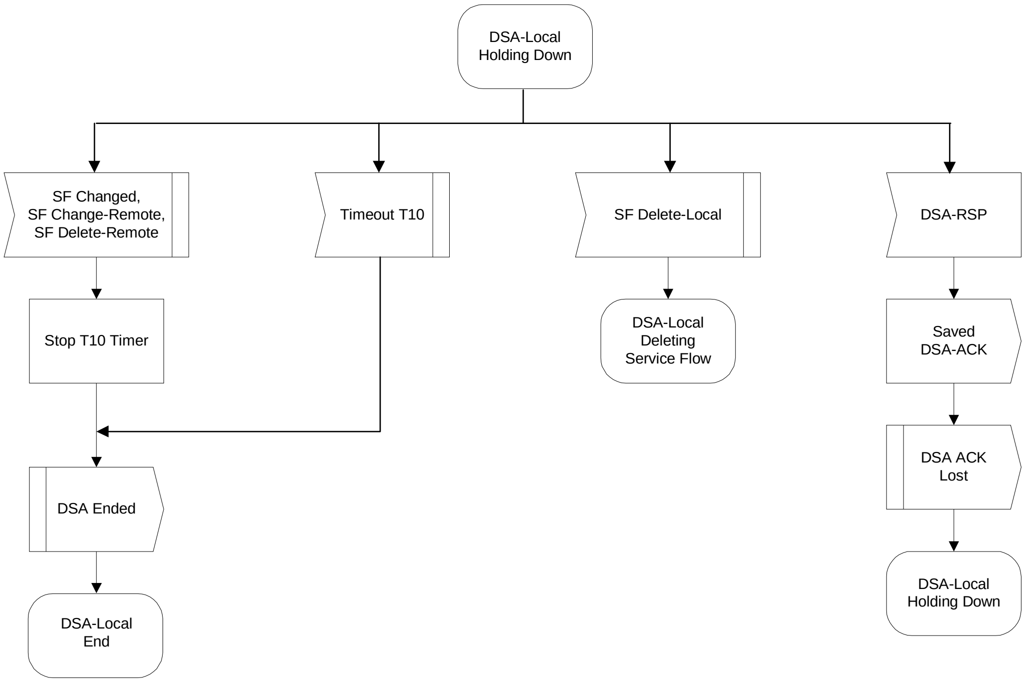


Figure 77—DSA—Locally initiated transaction holding state flow diagram

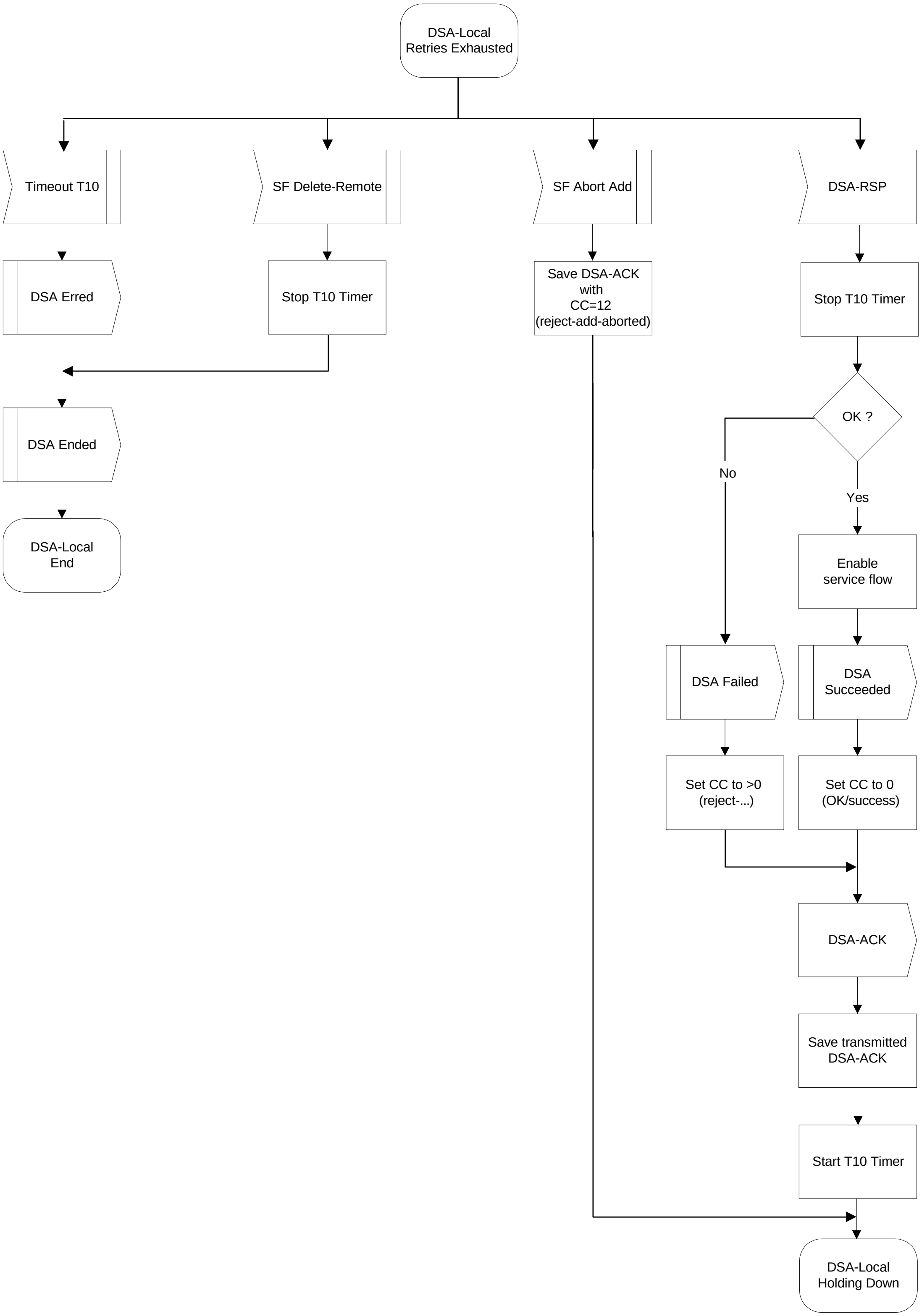


Figure 78—DSA—Locally initiated transaction retries exhausted state flow diagram

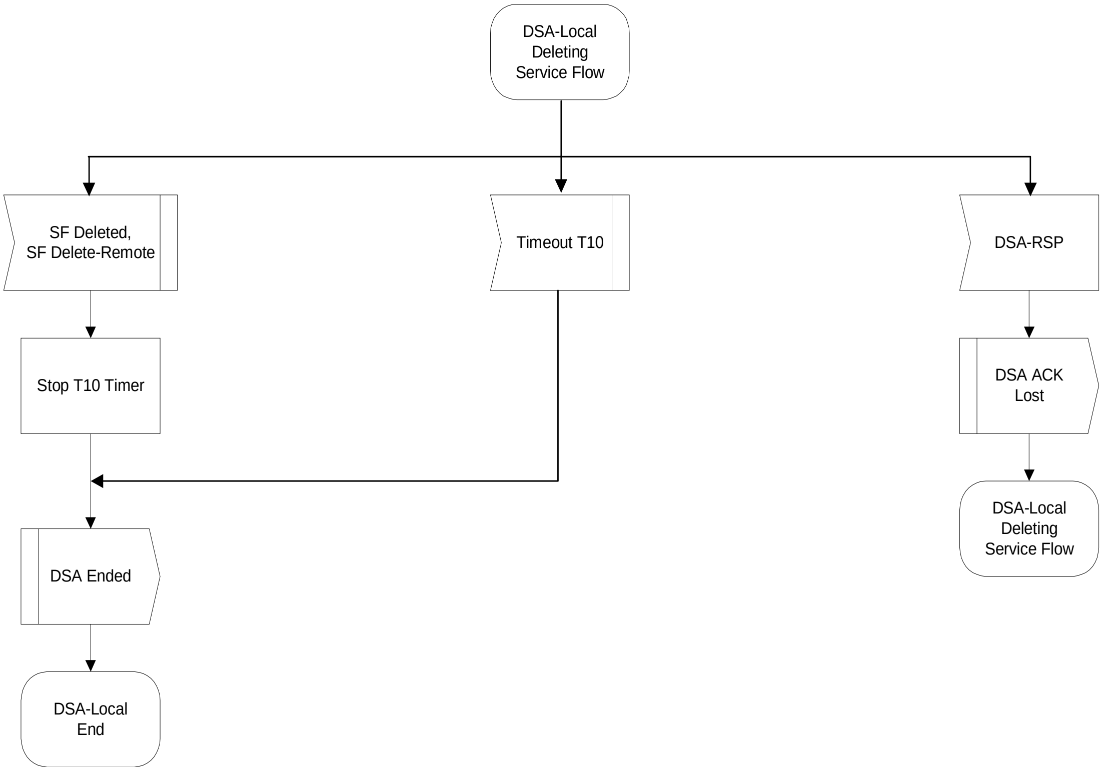


Figure 79—DSA—Locally initiated transaction deleting service flow state flow diagram

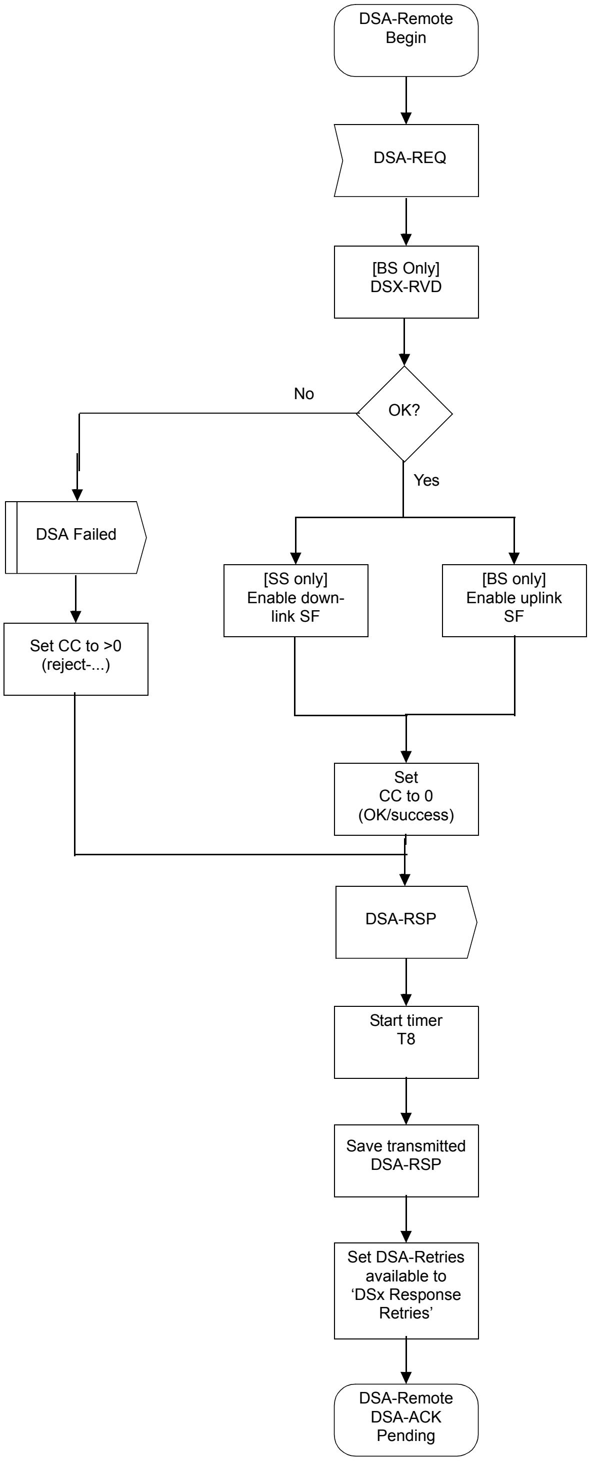

DSA-Remote Begin
DSA-REQ
[BS Only] DSX-RVD
OK?
No
Yes
DSA Failed
Set CC to >0 (reject-...)
[SS only] Enable down-link SF
[BS only] Enable uplink SF
Set CC to 0 (OK/success)
DSA-RSP
Start timer T8
Save transmitted DSA-RSP
Set DSA-Retries available to 'DSx Response Retries'
DSA-Remote DSA-ACK Pending

Figure 80—DSA—Remotely initiated transaction begin state flow diagram

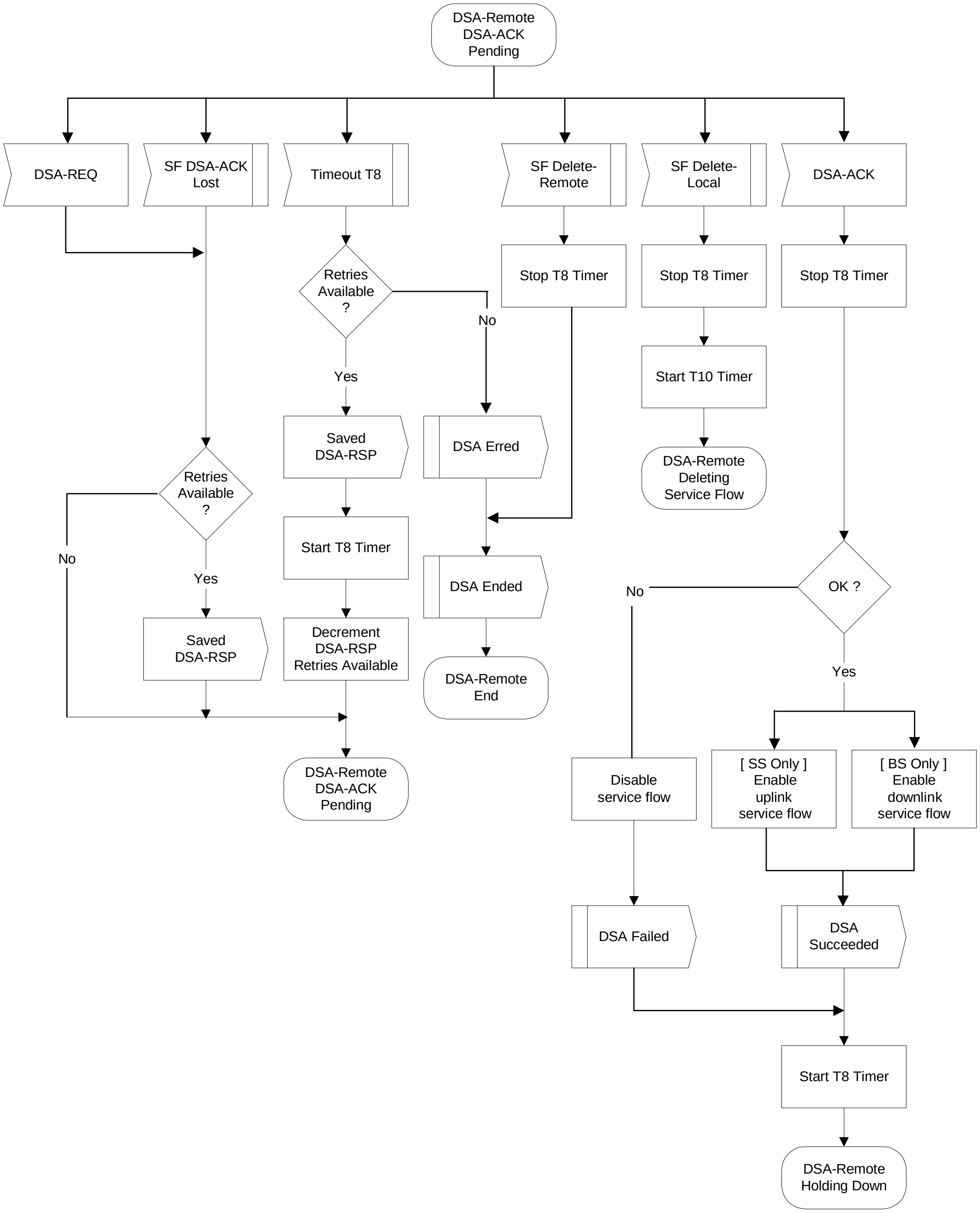


Figure 81—DSA—Remotely initiated transaction DSA-ACK pending state flow diagram

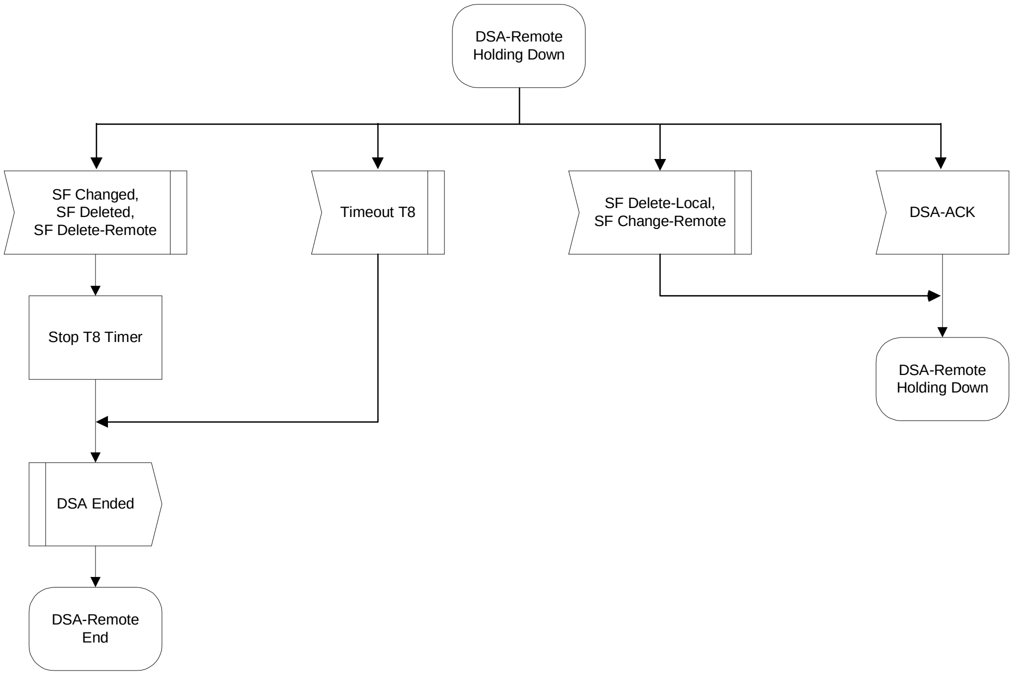


Figure 82—DSA—Remotely initiated transaction holding down state flow diagram

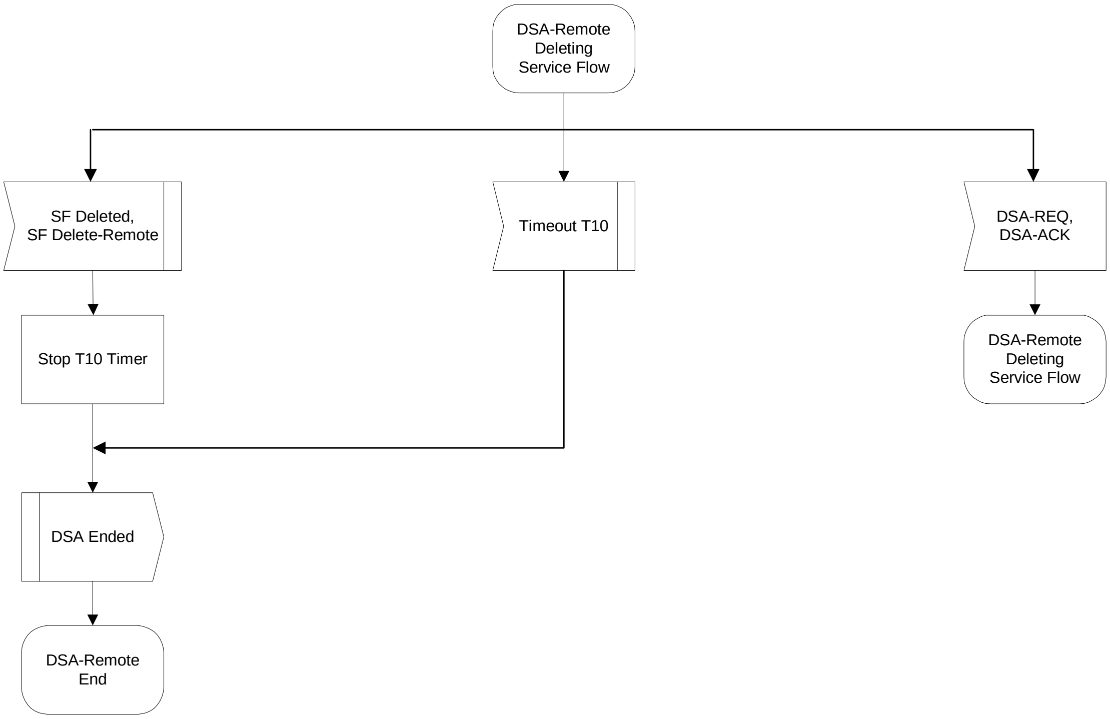

DSA-Remote Deleting Service Flow
SF Deleted, SF Delete-Remote
Timeout T10
DSA-REQ, DSA-ACK
Stop T10 Timer
DSA-Remote Deleting Service Flow
DSA Ended
DSA-Remote End

Figure 83—DSA—Remotely initiated transaction deleting service state flow diagram

6.2.13.8.4 Dynamic Service Change (DSC)

The DSC set of Messages is used to modify the flow parameters associated with a service flow. Specifically, DSC can modify the service flow Specification.

A single DSC Message exchange can modify the parameters of one downlink service flow and/or one uplink service flow.

To prevent packet loss, any required bandwidth change is sequenced between the SS and BS.

The BS controls both uplink and downlink scheduling. The timing of scheduling changes is independent of direction AND whether it's an increase or decrease in bandwidth. The BS always changes scheduling on receipt of a DSC-REQ (SS-initiated transaction) or DSC-RSP (BS-initiated transaction).

The BS also controls the downlink transmit behavior. The change in downlink transmit behavior is always coincident with the change in downlink scheduling (i.e., BS controls both and changes both simultaneously).

The SS controls the uplink transmit behavior. The timing of SS transmit behavior changes is a function of which device initiated the transaction AND whether the change is an "increase" or "decrease" in bandwidth.

If an uplink service flow's bandwidth is being reduced, the SS reduces its payload bandwidth first and then the BS reduces the bandwidth scheduled for the service flow. If an uplink service flow's bandwidth is being increased, the BS increases the bandwidth scheduled for the service flow first and then the SS increases its payload bandwidth.

Any service flow can be deactivated with a DSA command by sending a DSC-REQ Message, referencing the Service Flow Identifier, and including a null ActiveQoSParamSet. However, if a Basic, Primary Management, or Secondary Management Connection of an SS is deactivated that SS is deregistered and shall re-register. Therefore, care should be taken before deactivating such service flows. If a service flow that was provisioned during registration is deactivated, the provisioning information for that service flow shall be maintained until the service flow is reactivated.

An SS shall have only one DSC transaction outstanding per service flow. If it detects a second transaction initiated by the BS, the SS shall abort the transaction it initiated and allow the BS-initiated transaction to complete.

A BS shall have only one DSC transaction outstanding per service flow. If it detects a second transaction initiated by the SS, the BS shall abort the transaction that the SS initiated and allow the BS-initiated transaction to complete.

NOTE—Currently anticipated applications would probably control a service flow through either the SS or BS, and not both. Therefore the case of a DSC being initiated simultaneously by the SS and BS is considered as an exception condition and treated as one.

6.2.13.8.4.1 SS-initiated DSC

An SS that needs to change a service flow definition performs the following operations.

The SS informs the BS using a DSC-REQ. The BS checks the integrity of the message and, if the message is intact, sends a message received (DSX-RVD) response to the SS. The BS shall decide if the referenced service flow can support this modification. The BS shall respond with a DSC-RSP indicating acceptance or rejection. The SS reconfigures the service flow if appropriate, and then shall respond with a DSC-ACK. This process is illustrated in Table 68.

Table 68—SS-initiated DSC

BS		SS
		Service flow requires modifying
Receive DSC-REQ	<--------- DSC-REQ ----------	Send DSC-REQ Set Timers T7 and T14
DSC_REQ integrity valid	---------- DSX-RVD --------->	Timer T14 Stops
Validate Request		
Modify service flow		
Increase Channel Bandwidth if Required		
Send DSC-RSP	---------- DSC-RSP --------->	Receive DSC-RSP Timer T7 Stops
		Modify service flow
		Adjust Payload Bandwidth
Receive DSC-ACK	<--------- DSC-ACK ----------	Send DSC-ACK
Decrease Channel Bandwidth if Required		

6.2.13.8.4.2 BS-initiated DSC

A BS that needs to change a service flow definition performs the following operations.

The BS shall decide if the referenced service flow can support this modification. If so, the BS informs the SS using a DSC-REQ. The SS checks that it can support the service change, and shall respond using a DSC-RSP indicating acceptance or rejection. The BS reconfigures the service flow if appropriate, and then shall respond with a DSC-ACK. This process is illustrated in Table 69.

Table 69—BS-initiated DSC

BS		SS
Service flow requires modifying		
Send DSC-REQ Set Timer T7	---------- DSC-REQ --------->	Receive DSC-REQ
		Validate request
		Modify service flow
		Decrease Payload Bandwidth if Required
Receive DSC-RSP Timer T7 Stops	<--------- DSC-RSP ----------	Send DSC-RSP
Modify service flow		
Adjust Channel Bandwidth		
Send DSC-ACK	---------- DSC-ACK --------->	Receive DSC-ACK
		Increase Payload Bandwidth if Required

6.2.13.8.4.3 DSC state transition diagrams

DSC state transition diagrams are shown in Figure 84–Figure 92.

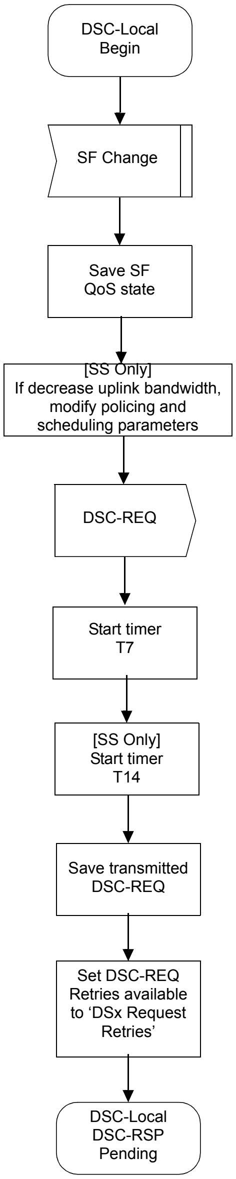

DSC-Local Begin
SF Change
Save SF QoS state
[SS Only] If decrease uplink bandwidth, modify policing and scheduling parameters
DSC-REQ
Start timer T7
[SS Only] Start timer T14
Save transmitted DSC-REQ
Set DSC-REQ Retries available to 'DSx Request Retries'
DSC-Local DSC-RSP Pending

Figure 84—DSC—Locally initiated transaction begin state flow diagram

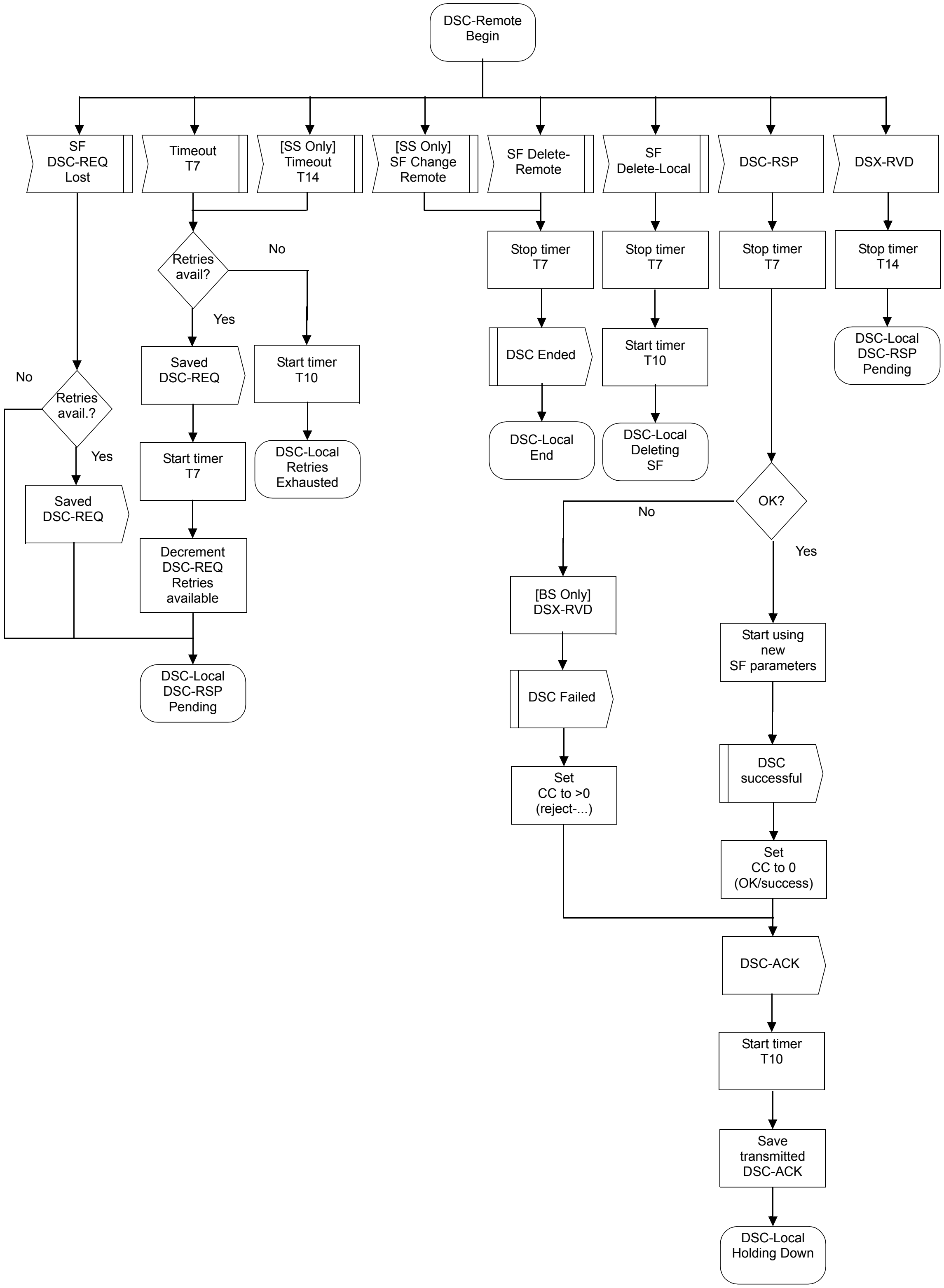


Figure 85—DSC—Locally initiated transaction DSC-RSP pending state flow diagram

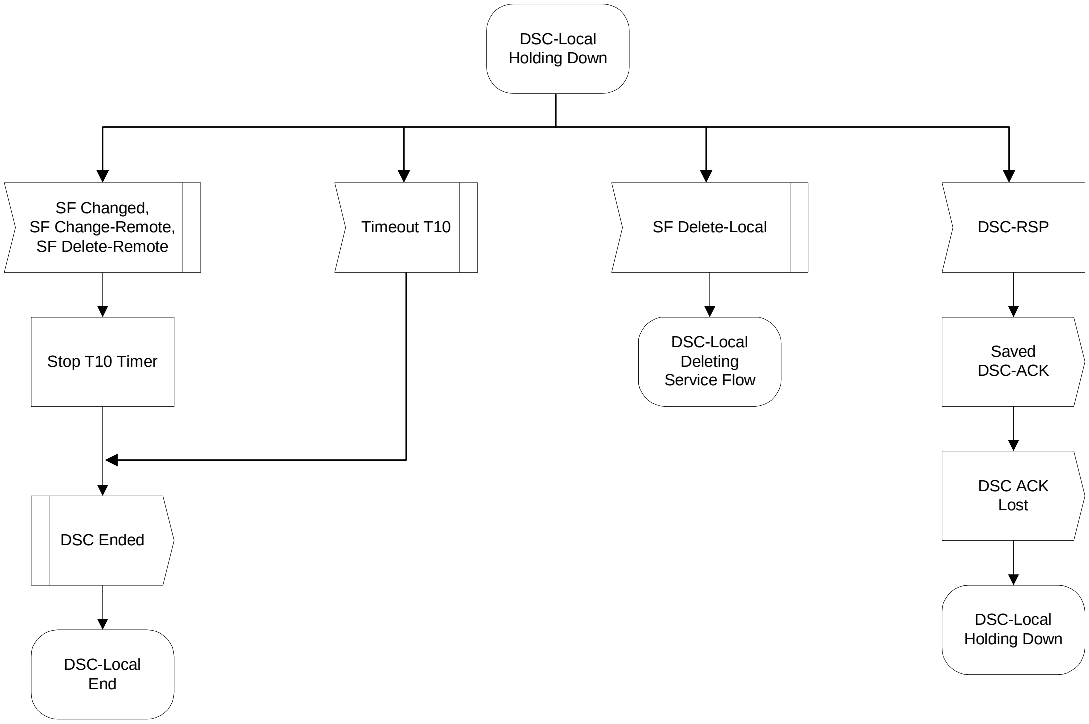


Figure 86—DSC—Locally initiated transaction holding down state flow diagram

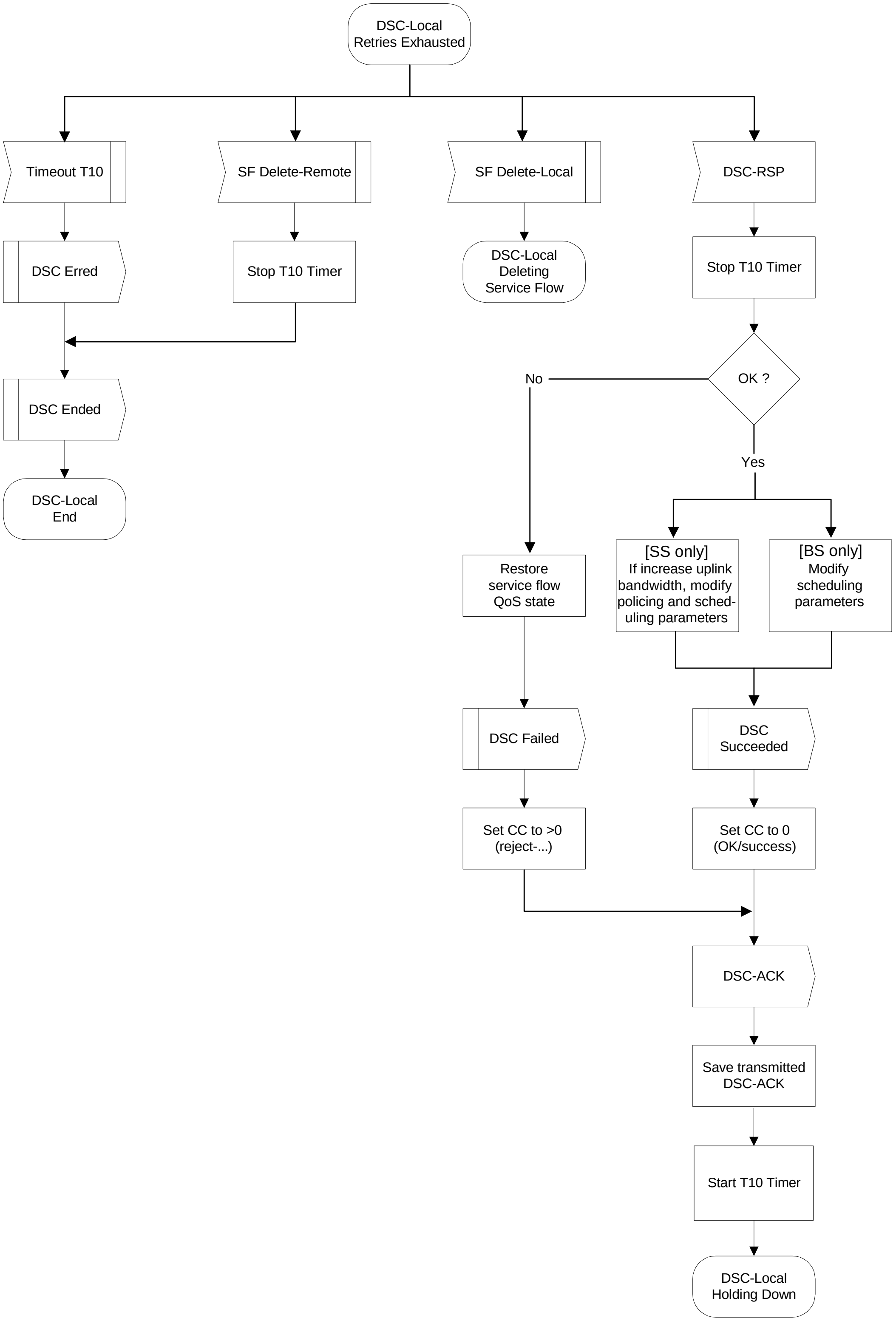


Figure 87—DSC—Locally initiated transaction retries exhausted state flow diagram

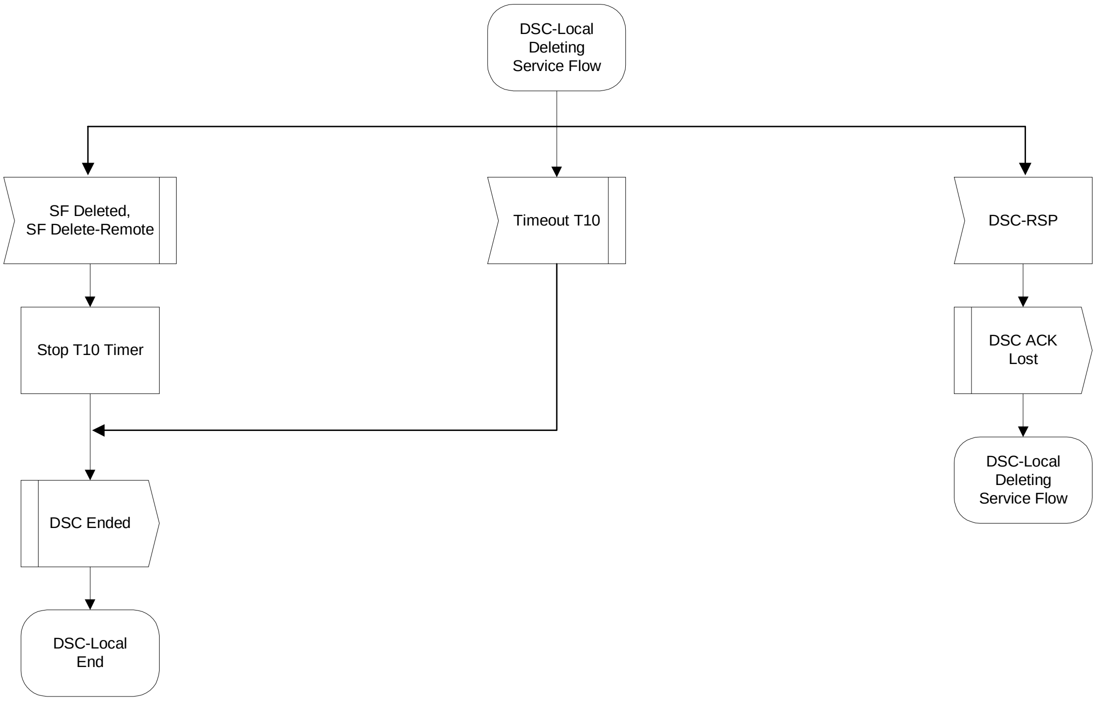


Figure 88—DSC—Locally initiated transaction deleting service flow state flow diagram

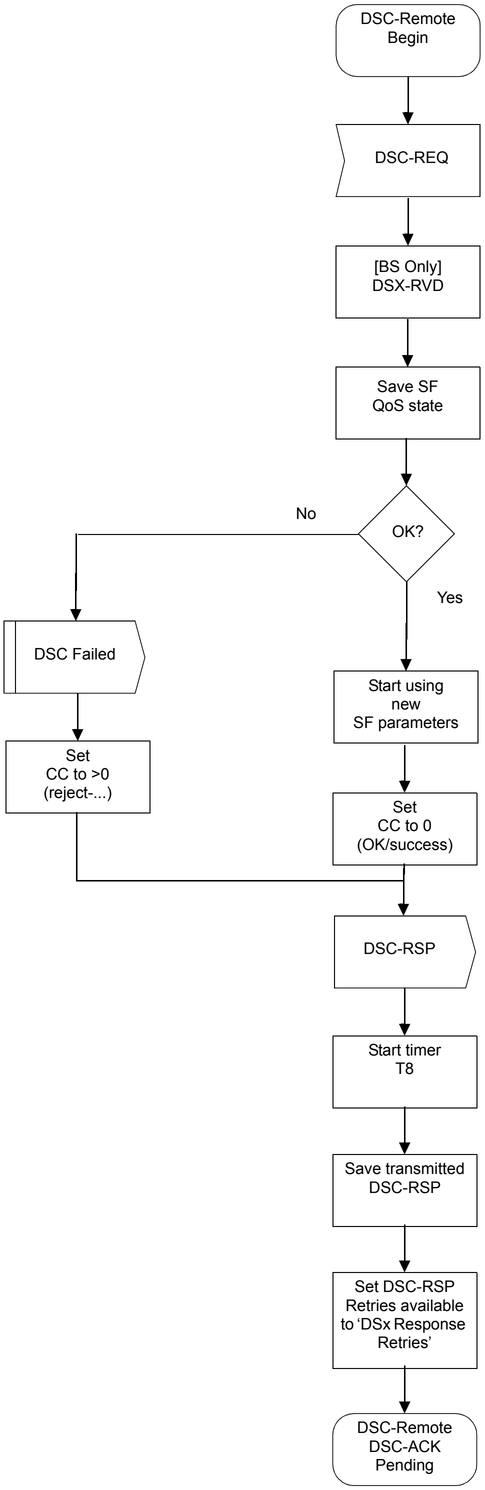


Figure 89—DSC—Remotely initiated transaction begin state flow diagram

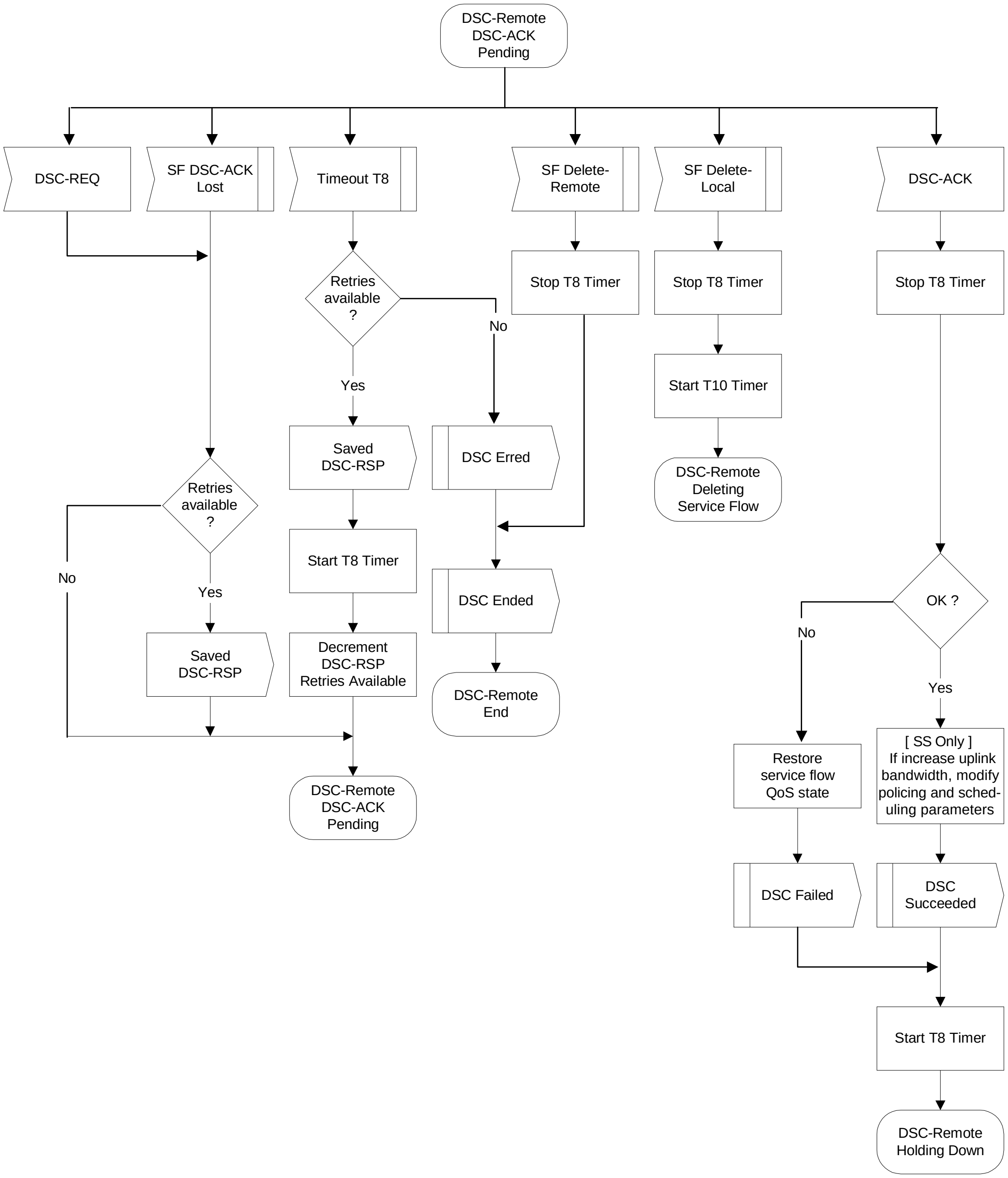


Figure 90—DSC—Remotely initiated transaction DSC-ACK pending state flow diagram

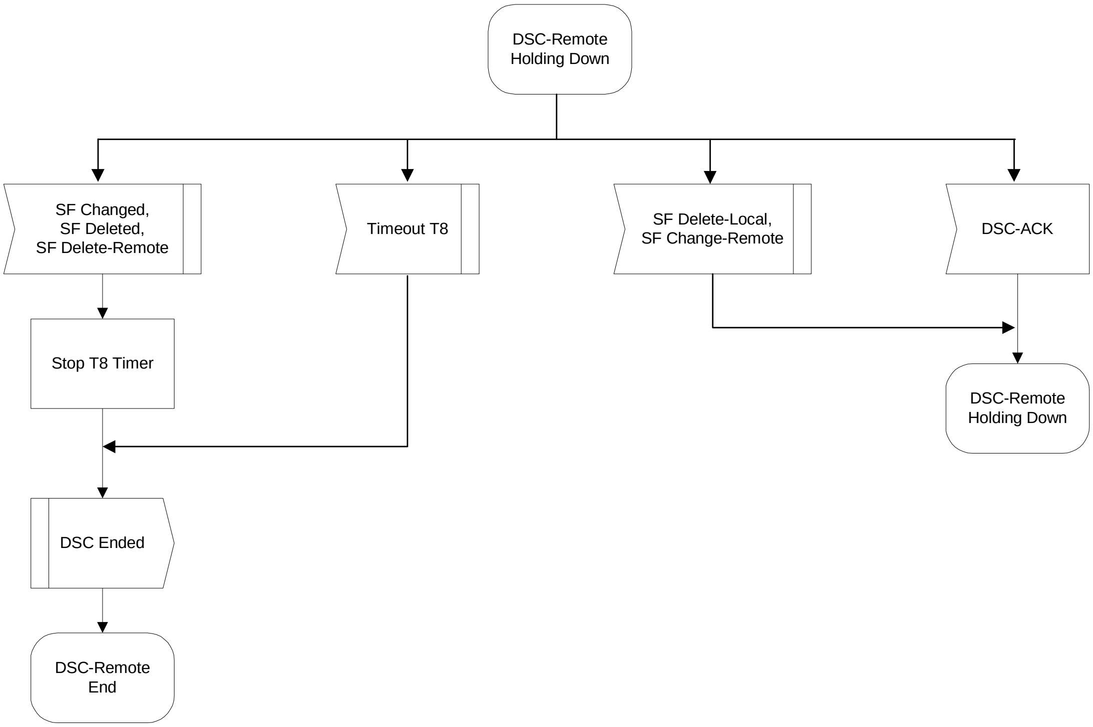


Figure 91—DSC—Remotely initiated transaction holding down state flow diagram

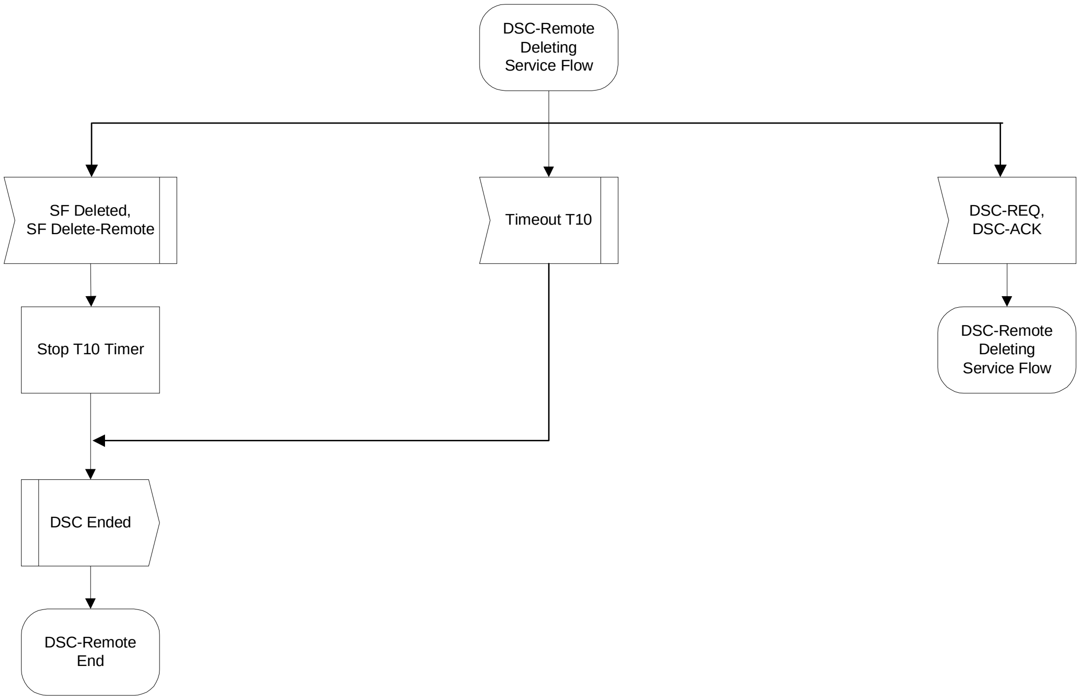


Figure 92—DSC—Remotely initiated transaction deleting service flow state flow diagram

6.2.13.8.5 Connection release

Any service flow can be deleted with the Dynamic Service Deletion (DSD) Messages. When a service flow is deleted, all resources associated with it are released. However, if a basic, primary management, or secondary management service flow of an SS is deleted, that SS is deregistered and shall reregister. Also, if a service flow that was provisioned during registration is deleted, the provisioning information for that service flow is lost until the SS reregisters. However, the deletion of a provisioned service flow shall not cause an SS to re-register. Therefore, care should be taken before deleting such service flows.

NOTE—Unlike DSA and DSC Messages, DSD Messages are limited to only a single service flow.

6.2.13.8.5.1 SS-initiated DSD

An SS wishing to delete a service flow generates a delete request to the BS using a Dynamic Service Deletion-Request Message (DSD-REQ). The BS removes the service flow and generates a response using a Dynamic Service Deletion-Response Message (DSD-RSP). This process is illustrated in Table 70. Only one service flow can be deleted per DSD-REQ.

Table 70—DSD-initiated from SS

SS		BS
Service flow no longer needed		
Delete service flow		
Send DSD-REQ	---DSD-REQ-->	Receive DSD-REQ
		Verify SS is service flow ‘owner’
		Delete fervice flow
Receive DSD-RSP	<--DSD-RSP---	Send DSD-RSP

6.2.13.8.5.2 BS-initiated DSD

A BS wishing to delete a dynamic service flow generates a delete request to the associated SS using a DSD-REQ. The SS removes the service flow and generates a response using a DSD-RSP. This process is illustrated in Table 71. Only one service flow can be deleted per DSD-REQ.

Table 71—DSD initiated from BS

SS		BS
		Service flow no longer needed
		Delete service flow
		Determine associated SS for this service flow
Receive DSD-REQ	<---DSD-REQ--	Send DSD-REQ
Delete service flow		
Send DSD-RSP	---DSD-RSP-->	Receive DSD-RSP

6.2.13.8.5.3 DSD state transition diagrams

DSD state transition diagrams are shown in Figure 93—Figure 96.

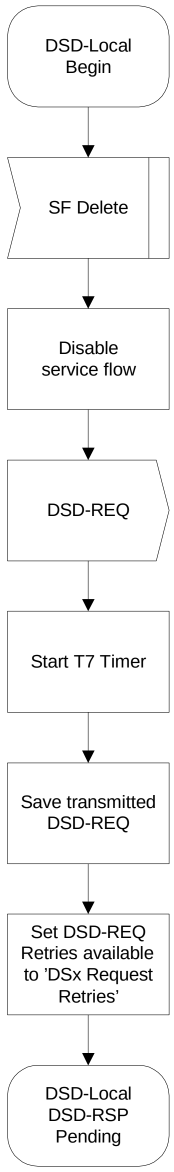


Figure 93—DSD—Locally initiated transaction begin state flow diagram

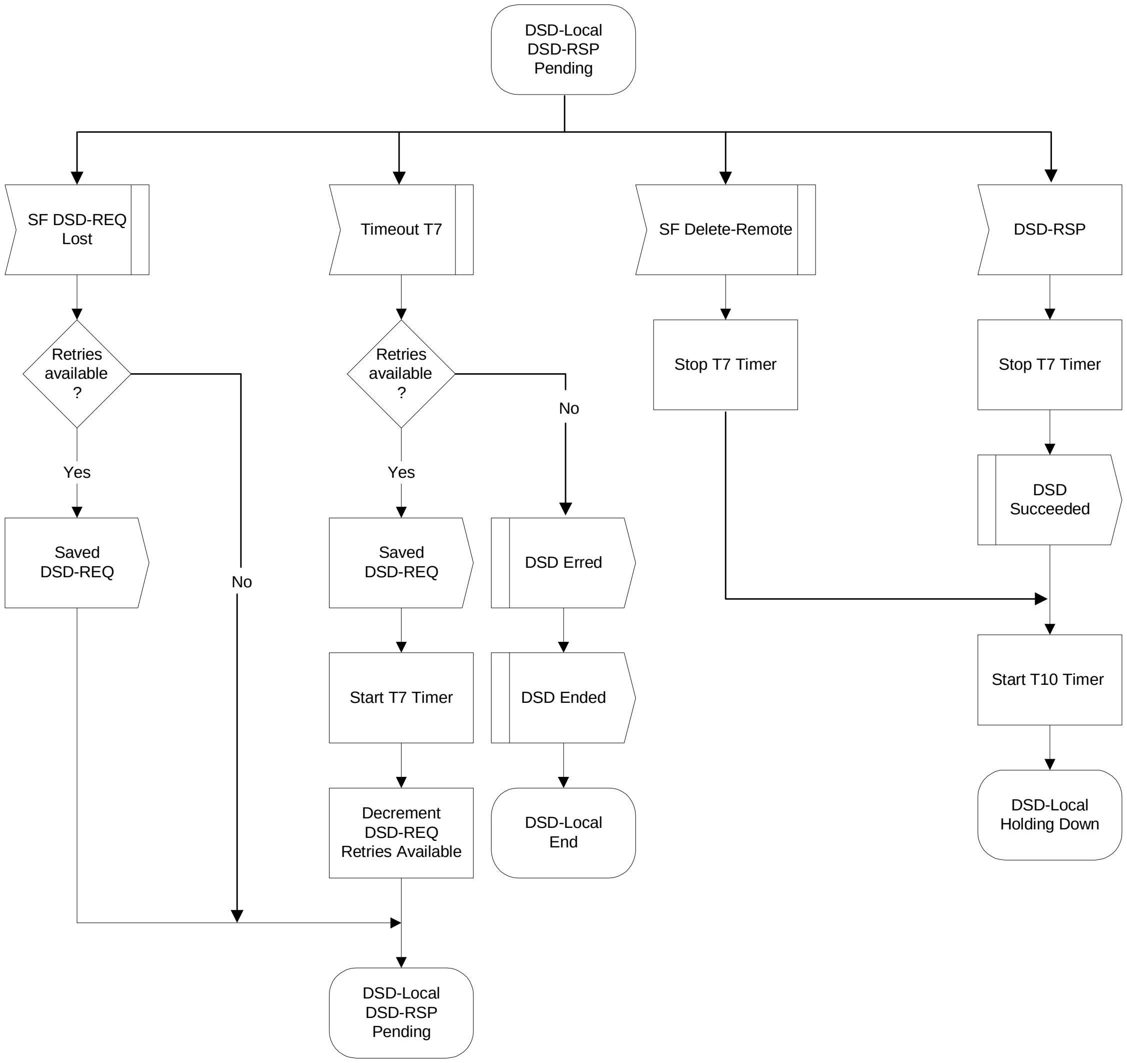


Figure 94—DSD—Locally Initiated transaction DSD-RSP pending state flow diagram

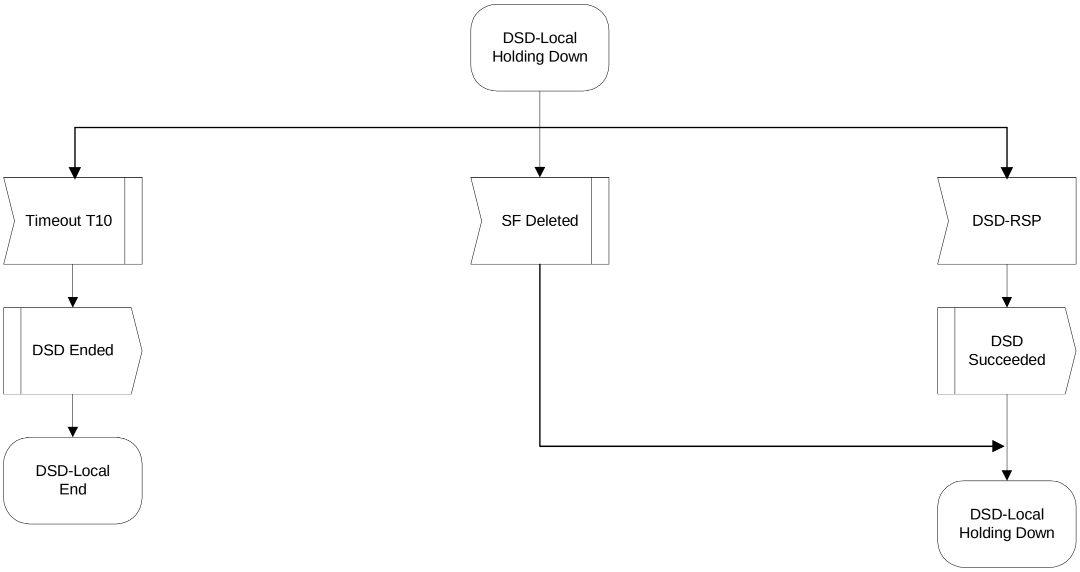


Figure 95—DSD—Locally initiated transaction holding down state flow diagram

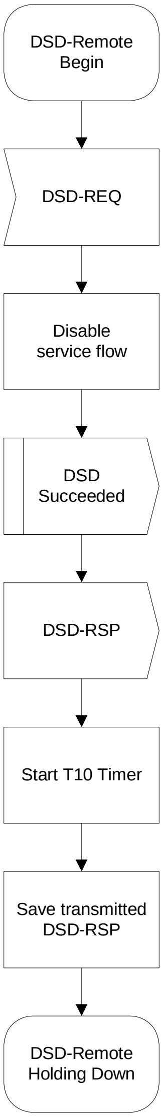


Figure 96—DSD—Remotely initiated transaction begin state flow diagram

7. Privacy sublayer

Privacy provides subscribers with privacy across the fixed broadband wireless network. It does this by encrypting connections between SS and BS.

In addition, Privacy provides operators with strong protection from theft of service. The BS protects against unauthorized access to these data transport services by enforcing encryption of the associated service flows across the network. Privacy employs an authenticated client/server key management protocol in which the BS, the server, controls distribution of keying material to client SS. Additionally, the basic privacy mechanisms are strengthened by adding digital-certificate -based SS authentication to its key management protocol.

7.1 Architecture

Privacy has two component protocols as follows:

a) An encapsulation protocol for encrypting packet data across the fixed broadband wireless access network. This protocol defines (1) a set of supported *cryptographic suites*, i.e., pairings of data encryption and authentication algorithms, and (2) the rules for applying those algorithms to a MAC PDU payload.

b) A key management protocol (Privacy Key Management, or PKM) providing the secure distribution of keying data from BS to SS. Through this key management protocol, SS and BS synchronize keying data; in addition, the BS uses the protocol to enforce conditional access to network services.

7.1.1 Packet data encryption

Encryption services are defined as a set of capabilities within the MAC Privacy Sublayer. MAC Header information specific to encryption is allocated in the Generic MAC Header Format.

Encryption is always applied to the MAC PDU payload; the Generic MAC Header is not encrypted. All MAC management messages shall be sent in the clear to facilitate registration, ranging, and normal operation of the MAC sublayer.

The format of MAC PDUs carrying encrypted packet data payloads is specified in 6.2.3.6.

7.1.2 Key management protocol

An SS uses the PKM protocol to obtain authorization and traffic keying material from the BS, and to support periodic reauthorization and key refresh. The key management protocol uses X.509 digital certificates [IETF RFC 2459], the RSA public-key encryption algorithm [PKCS #1], and strong symmetric algorithms to perform key exchanges between SS and BS.

The PKM protocol adheres to a client/server model, where the SS, a PKM "client," requests keying material, and the BS, a PKM "server," responds to those requests, ensuring that individual SS clients receive only keying material for which they are authorized. The PKM protocol uses MAC management messaging, i.e., PKM-REQ and PKM-RSP messages defined in 6.2.2.3.

The PKM protocol uses public-key cryptography to establish a shared secret (i.e., an Authorization Key) between SS and BS. The shared secret is then used to secure subsequent PKM exchanges of traffic encryption keys. This two-tiered mechanism for key distribution permits refreshing of traffic encryption keys without incurring the overhead of computation-intensive public-key operations.

A BS authenticates a client SS during the initial authorization exchange. Each SS carries a unique X.509 digital certificate issued by the SS's manufacturer. The digital certificate contains the SS's Public Key and SS MAC address. When requesting an Authorization Key, an SS presents its digital certificate to the BS. The BS verifies the digital certificate, and then uses the verified Public Key to encrypt an Authorization Key, which the BS then sends back to the requesting SS.

The BS associates an SS's authenticated identity to a paying subscriber, and hence to the data services that subscriber is authorized to access. Thus, with the Authorization Key exchange, the BS establishes an authenticated identity of a client SS and the services (i.e., specific traffic encryption keys) the SS is authorized to access.

Since the BS authenticates the SS, it can protect against an attacker employing a *cloned* SS, masquerading as a legitimate subscriber's SS. The use of the X.509 certificates prevents cloned SSs from passing fake credentials onto a BS.

All SSs shall have factory-installed RSA private/public key pairs or provide an internal algorithm to generate such key pairs dynamically. If an SS relies on an internal algorithm to generate its RSA key pair, the SS shall generate the key pair prior to its first Authorization Key (AK) exchange, described in 7.2.1. All SSs with factory-installed RSA key pairs shall also have factory-installed X.509 certificates. All SSs that rely on internal algorithms to generate an RSA key pair shall support a mechanism for installing a manufacturer-issued X.509 certificate following key generation.

The PKM protocol is defined in detail in 7.2.

7.1.3 Security Associations

A *Security Association* (SA) is the set of security information a BS and one or more of its client SS share in order to support secure communications across the IEEE Std 802.16-2001 network. Three types of SAs are defined: *Primary, Static*, and *Dynamic*. Each SS establishes a Primary Security association during the SS initialization process. Static SAs are provisioned within the BS. Dynamic SAs are established and eliminated, on the fly, in response to the initiation and termination of specific service flows. Both Static and Dynamic SAs can by shared by multiple SSs.

An SA's shared information shall include the Cryptographic Suite employed within the SA. The shared information may include Traffic Encryption Keys (TEKs) and Initialization Vectors. The exact content of the SA is dependent on the SA's Cryptographic Suite.

SAs are identified using SAIDs.

Each SS shall establish an exclusive Primary SA with its BS. The SAID of any SS's Primary SA shall be equal to the Basic CID of that SS.

Using the PKM protocol, an SS requests from its BS an SA's keying material. The BS shall ensure that each client SS only has access to the SAs it is authorized to access.

An SA's keying material [e.g., Data Encryption Standard (DES) key and CBC Initialization Vector] has a limited lifetime. When the BS delivers SA keying material to an SS, it also provides the SS with that material's remaining lifetime. It is the responsibility of the SS to request new keying material from the BS before the set of keying material that the SS currently holds expires at the BS. Should the current keying material expire before a new set of keying material is received, the SS shall perform network entry as described in 6.2.9. The PKM protocol specifies how SS and BS maintain key synchronization.

7.1.4 Mapping of connections to SAs

The following rules for mapping connections to SAs apply:

1) All Transport Connections shall be mapped to an existing SA.
2) Multicast Transport Connections may be mapped to any Static or Dynamic SA.
3) The Secondary Management Connection shall be mapped to the Primary SA.
4) The Basic and the Primary Management connections shall not be mapped to an SA.

The actual mapping is achieved by including the SAID of an existing SA in the DSA-xxx messages together with the CID. No explicit mapping of Secondary Management Connection to the Primary SA is required.

7.1.5 Cryptographic Suite

A Cryptographic Suite is the SA's set of methods for data encryption, data authentication, and TEK exchange. A Cryptographic Suite is specified as described in 11.2.14. The Cryptographic Suite shall be one of the ones listed in Table 137.

7.2 PKM protocol

7.2.1 SS authorization and AK exchange overview

SS authorization, controlled by the Authorization state machine, is the process of

a) the BS authenticating a client SS's identity

b) the BS providing the authenticated SS with an AK, from which a Key Encryption Key (KEK) and message authentication keys are derived

c) the BS providing the authenticated SS with the identities (i.e., the SAIDs) and properties of primary and static security associations the SS is authorized to obtain keying information for

After achieving initial authorization, an SS periodically seeks reauthorization with the BS; reauthorization is also managed by the SS's Authorization state machine. An SS must maintain its authorization status with the BS in order to be able to refresh aging TEKs. TEK state machines manage the refreshing of TEKs.

An SS begins authorization by sending an Authentication Information message to its BS. The Authentication Information message contains the SS manufacturer's X.509 certificate, issued by the manufacturer itself or by an external authority. The Authentication Information message is strictly informative; i.e., the BS may choose to ignore it. However, it does provide a mechanism for a BS to learn the manufacturer certificates of its client SS.

The SS sends an Authorization Request message to its BS immediately after sending the Authentication Information message. This is a request for an AK, as well as for the SAIDs identifying any Static Security SAs the SS is authorized to participate in. The Authorization Request includes

a) a manufacturer-issued X.509 certificate

b) a description of the cryptographic algorithms the requesting SS supports; an SS's cryptographic capabilities are presented to the BS as a list of cryptographic suite identifiers, each indicating a particular pairing of packet data encryption and packet data authentication algorithms the SS supports

c) the SS's Basic CID. The Basic CID is the first static CID the BS assigns to an SS during initial ranging—the primary SAID is equal to the Basic CID

In response to an Authorization Request message, a BS validates the requesting SS's identity, determines the encryption algorithm and protocol support it shares with the SS, activates an AK for the SS, encrypts it with the SS's public key, and sends it back to the SS in an Authorization Reply message. The authorization reply includes:

a) an AK encrypted with the SS's public key

b) a 4-bit key sequence number, used to distinguish between successive generations of AKs

c) a key lifetime

d) the identities (i.e., the SAIDs) and properties of the single primary and zero or more static SAs the SS is authorized to obtain keying information for

While the Authorization Reply shall identify Static SAs in addition to the Primary SA whose SAID matches the requesting SS's Basic CID, the Authorization Reply shall not identify any Dynamic SAs.

The BS, in responding to an SS's Authorization Request, shall determine whether the requesting SS, whose identity can be verified via the X.509 digital certificate, is authorized for basic unicast services, and what additional statically provisioned services (i.e., Static SAIDs) the SS's user has subscribed for. Note that the protected services a BS makes available to a client SS can depend upon the particular cryptographic suites SS and BS share support for.

An SS shall periodically refresh its AK by reissuing an Authorization Request to the BS. Reauthorization is identical to authorization with the exception that the SS does not send Authentication Information messages during reauthorization cycles. Subclause 7.2.4's description of the authorization state machine clearly indicates when Authentication Information messages are sent.

To avoid service interruptions during reauthorization, successive generations of the SS's AKs have overlapping lifetimes. Both SS and BS shall be able to support up to two simultaneously active AKs during these transition periods. The operation of the Authorization state machine's Authorization Request scheduling algorithm, combined with the BS's regimen for updating and using a client SS's Authorization Keys (see 7.4), ensures that the SS can refresh TEK keying information without interruption over the course of the SS's reauthorization periods.

7.2.2 TEK exchange overview

Upon achieving authorization, an SS starts a separate TEK state machine for each of the SAIDs identified in the Authorization Reply message. Each TEK state machine operating within the SS is responsible for managing the keying material associated with its respective SAID. TEK state machines periodically send Key Request messages to the BS, requesting a refresh of keying material for their respective SAIDs.

The BS responds to a Key Request with a Key Reply message, containing the BS's active keying material for a specific SAID.

The TEK in the Key Reply is triple DES (encrypt-decrypt-encrypt or EDE mode) encrypted, using a two-key, triple DES key encryption key (KEK) derived from the AK.

Note that at all times the BS maintains two active sets of keying material per SAID. The lifetimes of the two generations overlap such that each generation becomes active halfway through the life of it predecessor and expires halfway through the life of its successor. A BS includes in its Key Replies *both* of an SAID's active generations of keying material.

The Key Reply provides the requesting SS, in addition to the TEK and CBC initialization vector, the remaining lifetime of each of the two sets of keying material. The receiving SS uses these remaining lifetimes to estimate when the BS will invalidate a particular TEK, and therefore when to schedule future Key Requests such that the SS requests and receives new keying material before the BS expires the keying material the SS currently holds.

The operation of the TEK state machine's Key Request scheduling algorithm, combined with the BS's regimen for updating and using an SAID's keying material (see 7.4), ensures that the SS will be able to continually exchange encrypted traffic with the BS.

A TEK state machine remains active as long as

a) the SS is authorized to operate in the BS's security domain, i.e., it has a valid AK, and

b) the SS is authorized to participate in that particular SA, i.e., the BS continues to provide fresh keying material during rekey cycles.

The parent Authorization state machine stops *all* of its child TEK state machines when the SS receives from the BS an Authorization Reject during a reauthorization cycle. Individual TEK state machines can be started or stopped during a reauthorization cycle if an SS's Static SAID authorizations changed between successive re-authorizations.

Communication between Authorization and TEK state machines occurs through the passing of events and protocol messaging. The Authorization state machine generates events (i.e., Stop, Authorized, Authorization Pending, and Authorization Complete events) that are targeted at its child TEK state machines. TEK state machines do not target events at their parent Authorization state machine. The TEK state machine affects the Authorization state machine indirectly through the messaging a BS sends in response to an SS's requests: a BS may respond to a TEK machine's Key Requests with a failure response (i.e., Authorization Invalid message) to be handled by the Authorization state machine.

7.2.3 Security capabilities selection

As part of their authorization exchange, the SS provides the BS with a list of all the cryptographic suites (pairing of data encryption and data authentication algorithms) the SS supports. The BS selects from this list a single cryptographic suite to employ with the requesting SS's primary SA. The Authorization Reply the BS sends back to the SS includes a primary SA descriptor which, among other things, identifies the cryptographic suite the BS selected to use for the SS's primary SA. A BS shall reject the authorization request if it determines that none of the offered cryptographic suites are satisfactory.

The Authorization Reply also contains an optional list of static SA descriptors; each static SA descriptor identifies the cryptographic suite employed within the SA. The selection of a static SA's cryptographic suite is typically made independent of the requesting SS's cryptographic capabilities. A BS may include in its Authorization Reply static SA descriptors identifying cryptographic suites the requesting SS does not support; if this is the case, the SS shall not start TEK state machines for static SAs whose cryptographic suites the SS does not support.

7.2.4 Authorization state machine

The Authorization state machine consists of six states and eight distinct events (including receipt of messages) that can trigger state transitions. The Authorization finite state machine (FSM) is presented below in a graphical format, as a state flow model (Figure 97), and in a tabular format, as a state transition matrix (Table 72).

The state flow diagram depicts the protocol messages transmitted and internal events generated for each of the model's state transitions; however, the diagram does not indicate additional internal actions, such as the clearing or starting of timers, that accompany the specific state transitions. Accompanying the state transition matrix is a detailed description of the specific actions accompanying each state transition; the state transition matrix shall be used as the definitive specification of protocol actions associated with each state transition.

The following legend applies to the Authorization State Machine flow diagram depicted in Figure 97.

a) Ovals are states.

b) Events are in *italics*.

c) Messages are in normal font.

d) State transitions (i.e., the lines between states) are labeled with <what causes the transition>/<messages and events triggered by the transition>. So "*timeout*/Auth Request" means that the state received a "timeout" event and sent an Authorization Request ("Auth Request") message. If there are multiple events or messages before the slash "/" separated by a comma, *any* of them can cause the transition. If there are multiple events or messages listed after the slash, *all* of the specified actions shall accompany the transition.

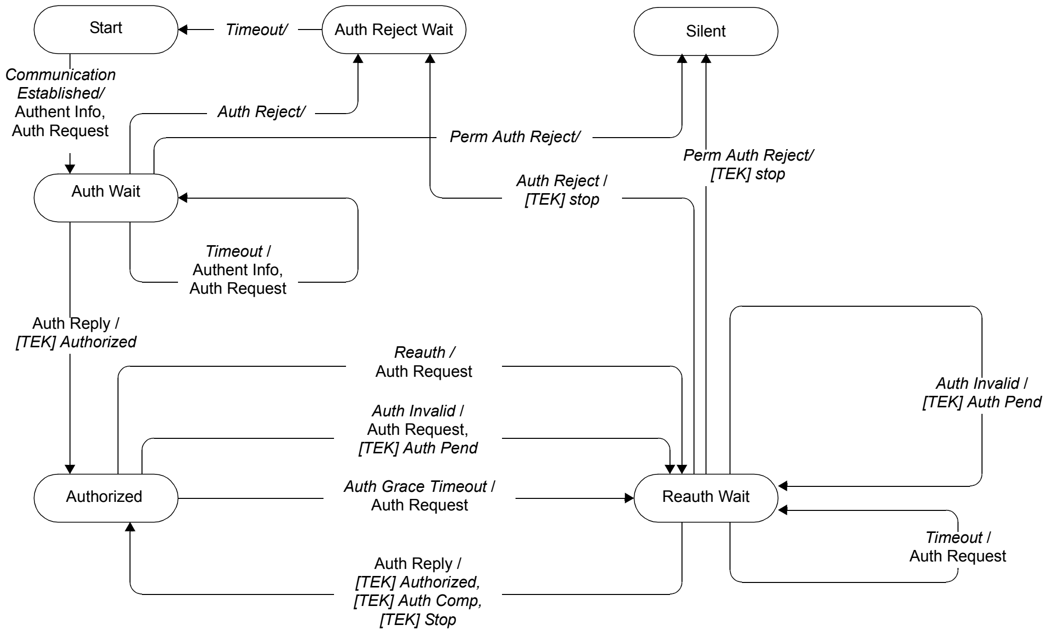


Figure 97—Authorization state machine flow diagram

Table 72—Authorization FSM state transition matrix

State ***Event or Rcvd Message***	**(A) Start**	**(B) Auth Wait**	**(C) Authorized**	**(D) Reauth Wait**	**(E) Auth Reject Wait**	**(F) Silent**
(1) *Communication Established*	Auth Wait					
(2) *Auth Reject*		Auth Reject Wait		Auth Reject Wait		
(3) *Perm Auth Reject*		Silent		Silent		
(4) Auth Reply		Authorized		Authorized		
(5) *Timeout*		Auth Wait		Reauth Wait	Start	
(6) *Auth Grace Timeout*			Reauth Wait			
(7) *Auth Invalid*			Reauth Wait	Reauth Wait		
(8) *Reauth*			Reauth Wait			

The Authorization state transition matrix presented in Table 72 lists the six Authorization machine states in the topmost row and the eight Authorization machine events (includes message receipts) in the leftmost column. Any cell within the matrix represents a specific combination of state and event, with the next state (the state transitioned to) displayed within the cell. For example, cell 4-B represents the receipt of an Authorization Reply (Auth Reply) message when in the Authorize Wait (Auth Wait) state. Within cell 4-B is the name of the next state, "Authorized." Thus, when an SS's Authorization state machine is in the Auth Wait state and an Auth Reply message is received, the Authorization state machine will transition to the Authorized state. In conjunction with this state transition, several protocol actions shall be taken; these are described in the listing of protocol actions, under the heading 4-B, in 7.2.4.5.

A shaded cell within the state transition matrix implies that either the specific event cannot or should not occur within that state, and if the event does occur, the state machine shall ignore it. For example, if an Auth Reply message arrives when in the Authorized state, that message should be ignored (cell 4-C). The SS may, however, in response to an improper event, log its occurrence, generate an SNMP event, or take some other vendor-defined action. These actions, however, are not specified within the context of the Authorization state machine, which simply ignores improper events.

7.2.4.1 States

a) *Start:* This is the initial state of the FSM. No resources are assigned to or used by the FSM in this state—e.g., all timers are off, and no processing is scheduled.

b) *Authorize Wait (Auth Wait):* The SS has received the "Communication Established" event indicating that it has completed basic capabilities negotiation with the BS. In response to receiving the event, the SS has sent both an Authentication Information and an Auth Request message to the BS and is waiting for the reply.

c) *Authorized:* The SS has received an Auth Reply message which contains a list of valid SAIDs for this SS. At this point, the SS has a valid AK and SAID list. Transition into this state triggers the creation of one TEK FSM for each of the SS's privacy-enabled SAIDs.

d) *Reauthorize Wait (Reauth Wait):* The SS has an outstanding reauthorization request. The SS was either about to expire (see Authorization Grace Time in Table 119) its current authorization or received an indication (an Authorization Invalid message from the BS) that its authorization is no longer valid. The SS sent an Auth Request message to the BS and is waiting for a response.

e) *Authorize Reject Wait (Auth Reject Wait):* The SS received an Authorization Reject (Auth Reject) message in response to its last Auth Request. The Auth Reject's error code indicated the error was not of a permanent nature. In response to receiving this reject message, the SS set a timer and transitioned to the Auth Reject Wait state. The SS remains in this state until the timer expires.

f) *Silent:* The SS received an Auth Reject message in response to its last Auth Request. The Auth Reject's error code indicated the error was of a permanent nature. This triggers a transition to the Silent state, where the SS is not permitted to pass subscriber traffic. The SS shall, however, respond to management messages from the BS issuing the Perm Auth Reject.

7.2.4.2 Messages

Note that the message formats are defined in detail in 6.2.2.3.9.

Authorization Request (Auth Request): Request an AK and list of authorized SAIDs. Sent from SS to BS.

Authorization Reply (Auth Reply): Receive an AK and list of authorized, static SAIDs. Sent from BS to SS. The Authorization Key is encrypted with the SS's public key.

Authorization Reject (Auth Reject): Attempt to authorize was rejected. Sent from the BS to the SS.

Authorization Invalid (Auth Invalid): The BS may send an Authorization Invalid message to a client SS as follows:

a) an unsolicited indication, or

b) a response to a message received from that SS.

In either case, the Auth Invalid message instructs the receiving SS to re-authorize with its BS.

The BS responds to a Key Request with an Auth Invalid message if (1) the BS does not recognize the SS as being authorized (i.e., no valid AK associated with SS) or (2) verification of the Key Request's keyed message digest (in HMAC-Digest Attribute) failed. Note that the Authorization Invalid *event*, referenced in both the state flow diagram and the state transition matrix, signifies either the receipt of an Auth Invalid message or an internally generated event.

Authentication Information (Authent Info): The Auth Info message contains the SS manufacturer's X.509 Certificate, issued by an external authority. The Authent Info message is strictly an informative message the SS sends to the BS; with it, a BS may dynamically learn the manufacturer certificate of client SS. Alternatively, a BS may require out-of-band configuration of its list of manufacturer certificates.

7.2.4.3 Events

Communication Established: The Authorization state machine generates this event upon entering the Start state if the MAC has completed basic capabilities negotiation. If the basic capabilities negotiation is not complete, the SS sends a Communication Established event to the Authorization FSM upon completing basic capabilities negotiation. The Communication Established event triggers the SS to begin the process of getting its AK and TEKs.

Timeout: A retransmission or wait timer timed out. Generally a request is resent.

Authorization Grace Timeout (Auth Grace Timeout): The Authorization Grace timer timed out. This timer fires a configurable amount of time (the Authorization Grace Time) before the current authorization is supposed to expire, signalling the SS to reauthorize before its authorization actually expires. The Authorization Grace Time is specified in a configuration setting within the TFTP-downloaded SS Configuration File (9.2).

Reauthorize (Reauth): SS's set of authorized static SAIDs may have changed. This event is generated in response to an SNMP set and meant to trigger a reauthorization cycle.

Authorization Invalid (Auth Invalid): This event is internally generated by the SS when there is a failure authenticating a Key Reply or Key Reject message, or externally generated by the receipt of an Auth Invalid message, sent from the BS to the SS. A BS responds to a Key Request with an Auth Invalid if verification of the request's message authentication code fails. Both cases indicate BS and SS have lost AK synchronization.

A BS may also send to an SS an unsolicited Auth Invalid message, forcing an Auth Invalid event.

Permanent Authorization Reject (Perm Auth Reject): The SS receives an Auth Reject in response to an Auth Request. The error code in the Auth Reject indicates the error is of a permanent nature. What is interpreted as a permanent error is subject to administrative control within the BS. Auth Request processing errors that can be interpreted as permanent error conditions include:

a) unknown manufacturer (do not have CA certificate of the issuer of the SS Certificate)

b) invalid signature on SS certificate

c) ASN.1 parsing failure

d) inconsistencies between data in the certificate and data in accompanying PKM data Attributes

e) incompatible security capabilities

When an SS receives an Auth Reject indicating a permanent failure condition, the Authorization State machine moves into a Silent state, where the SS is not permitted to pass subscriber traffic. The SS shall, however, respond to management messages from the BS issuing the Perm Auth Reject. The SS shall also issue an SNMP Trap upon entering the Silent state.

Authorization Reject (Auth Reject): The SS receives an Auth Reject in response to an Auth Request. The error code in the Auth Reject does not indicate the failure was due to a permanent error condition. As a result, the SS's Authorization state machine shall set a wait timer and transition into the Auth Reject Wait State. The SS shall remain in this state until the timer expires, at which time it shall reattempt authorization.

NOTE—The following events are sent by an Authorization state machine to the TEK state machine:

[TEK] Stop: Sent by the Authorization FSM to an active (non-START state) TEK FSM to terminate the FSM and remove the corresponding SAID's keying material from the SS's key table.

[TEK] Authorized: Sent by the Authorization FSM to a non-active (START state), but valid TEK FSM.

[TEK] Authorization Pending (Auth Pend): Sent by the Authorization FSM to a specific TEK FSM to place that TEK FSM in a wait state until the Authorization FSM can complete its reauthorization operation.

[TEK] Authorization Complete (Auth Comp): Sent by the Authorization FSM to a TEK FSM in the Operational Reauthorize Wait (Op Reauth Wait) or Rekey Reauthorize Wait (Rekey Reauth Wait) states to clear the wait state begun by a TEK FSM Authorization Pending event.

7.2.4.4 Parameters

All configuration parameter values are specified in the TFTP-downloaded SS Configuration File (see 9.2).

Authorize Wait Timeout (Auth Wait Timeout): Timeout period between sending Authorization Request messages from Auth Wait state (see 11.2.19.5).

Authorization Grace Timeout (Auth Grace Timeout): Amount of time before authorization is scheduled to expire that the SS starts reauthorization (see 11.2.19.3.

Authorize Reject Wait Timeout (Auth Reject Wait Timeout): Amount of time an SS's Authorization FSM remains in the Auth Reject Wait state before transitioning to the Start state (see 11.2.19.7).

7.2.4.5 Actions

Actions taken in association with state transitions are listed by <event> (<rcvd message>) --> <state> below:

1-A Start (*Communication Established*) → Auth Wait

a) send Authent Info message to BS

b) send Auth Request message to BS

c) set Auth Request retry timer to Auth Wait Timeout

2-B Auth Wait (*Auth Reject*) → Auth Reject Wait

a) clear Auth Request retry timer

b) set a wait timer to Auth Reject Wait Timeout

2-D Reauth Wait (*Auth Reject*) → Auth Reject Wait

a) clear Auth Request retry timer

b) generate TEK FSM Stop events for all active TEK state machines

c) set a wait timer to Auth Reject Wait Timeout

3-B Auth Wait (*Perm Auth Reject*) → Silent

a) clear Auth Request retry timer

b) disable all forwarding of SS traffic

3-D Reauth Wait (*Perm Auth Reject*) → Silent

a) clear Auth Request retry timer

b) generate TEK FSM Stop events for all active TEK state machines

c) disable all forwarding of SS traffic

4-B Auth Wait (Auth Reply) → Authorized

a) clear Auth Request retry timer

b) decrypt and record AK delivered with Auth Reply

c) start TEK FSMs for all SAIDs listed in Authorization Reply (provided the SS supports the cryptographic suite that is associated with an SAID) and issue a TEK FSM Authorized event for each of the new TEK FSMs

d) set the Authorization Grace timer to go off “Authorization Grace Time” seconds prior to the supplied AK’s scheduled expiration

4-D Reauth Wait (Auth Reply) → Authorized

a) clear Auth Request retry timer

b) decrypt and record AK delivered with Auth Reply

c) start TEK FSMs for any newly authorized SAIDs listed in Auth Reply (provided the SS supports the cryptographic suite that is associated with the new SAID) and issue TEK FSM Authorized event for each of the new TEK FSMs

d) generate TEK FSM Authorization Complete events for any currently active TEK FSMs whose corresponding SAIDs were listed in Auth Reply

e) generate TEK FSM Stop events for any currently active TEK FSMs whose corresponding SAIDs were not listed in Auth Reply

f) set the Authorization Grace timer to go off “Authorization Grace Time” seconds prior to the supplied AK’s scheduled expiration

<u>5-B</u> Auth Wait (*Timeout*) → Auth Wait

a) send Authent Info message to BS
b) send Auth Request message to BS
c) set Auth Request retry timer to Auth Wait Timeout

<u>5-D</u> Reauth Wait (*Timeout*) → Reauth Wait

a) send Auth Request message to BS
b) set Auth Request retry timer to Reauth Wait Timeout

<u>5-E</u> Auth Reject Wait (*Timeout*) → Start

a) no protocol actions associated with state transition

<u>6-C</u> Authorized (*Auth Grace Timeout*) → Reauth Wait

a) send Auth Request message to BS
b) set Auth Request retry timer to Reauth Wait Timeout

<u>7-C</u> Authorized (*Auth Invalid*) → Reauth Wait

a) clear Authorization Grace timer
b) send Auth Request message to BS
c) set Auth Request retry timer to Reauth Wait Timeout
d) if the Auth Invalid event is associated with a particular TEK FSM, generate a TEK FSM Authorization Pending event for the TEK state machine responsible for the Auth Invalid event (i.e., the TEK FSM that either generated the event, or sent the Key Request message the BS responded to with an Auth Invalid message)

<u>7-D</u> Reauth Wait (*Auth Invalid*) → Reauth Wait

a) if the Auth Invalid event is associated with a particular TEK FSM, generate a TEK FSM Authorization Pending event for the TEK state machine responsible for the Auth Invalid event (i.e., the TEK FSM that either generated the event, or sent the Key Request message the BS responded to with an Auth Invalid message)

<u>8-C</u> Authorized (*Reauth*) → Reauth Wait

a) clear Authorization Grace timer
b) send Auth Request message to BS
c) set Auth Request retry timer to Reauth Wait Timeout

7.2.5 TEK state machine

The TEK state machine consists of six states and nine events (including receipt of messages) that can trigger state transitions. Like the Authorization state machine, the TEK state machine is presented in both a state flow diagram (Figure 98) and a state transition matrix (Table 73). As was the case for the Authorization state machine, the state transition matrix shall be used as the definitive specification of protocol actions associated with each state transition.

Shaded states in Figure 98 (Operational, Rekey Wait, and Rekey Reauthorize Wait) have valid keying material and encrypted traffic can be passed.

The Authorization state machine starts an independent TEK state machine for each of its authorized SAIDs.

As mentioned in 7.2.2, the BS maintains two active TEKs per SAID. The BS includes in its Key Replies both of these TEKs, along with their remaining lifetimes. The BS encrypts downlink traffic with the older of its two TEKs and decrypts uplink traffic with either the older or newer TEK, depending upon which of the two keys the SS was using at the time. The SS encrypts uplink traffic with the newer of its two TEKs and decrypts downlink traffic with either the older or newer TEK, depending upon which of the two keys the BS was using at the time. See 7.4 for details on SS and BS key usage requirements.

Through operation of a TEK state machine, the SS attempts to keep its copies of an SAID’s TEKs synchronized with those of its BS. A TEK state machine issues Key Requests to refresh copies of its SAID’s keying material soon after the scheduled expiration time of the older of its two TEKs and before the expiration of its newer TEK. To accommodate for SS/BS clock skew and other system processing and transmission delays, the SS schedules its Key Requests a configurable number of seconds before the newer TEK’s estimated expiration in the BS. With the receipt of the Key Reply, the SS shall always update its records with the TEK Parameters from both TEKs contained in the Key Reply Message. Figure 98 illustrates the SS’s scheduling of its key refreshes in conjunction with its management of an SA’s active TEKs.

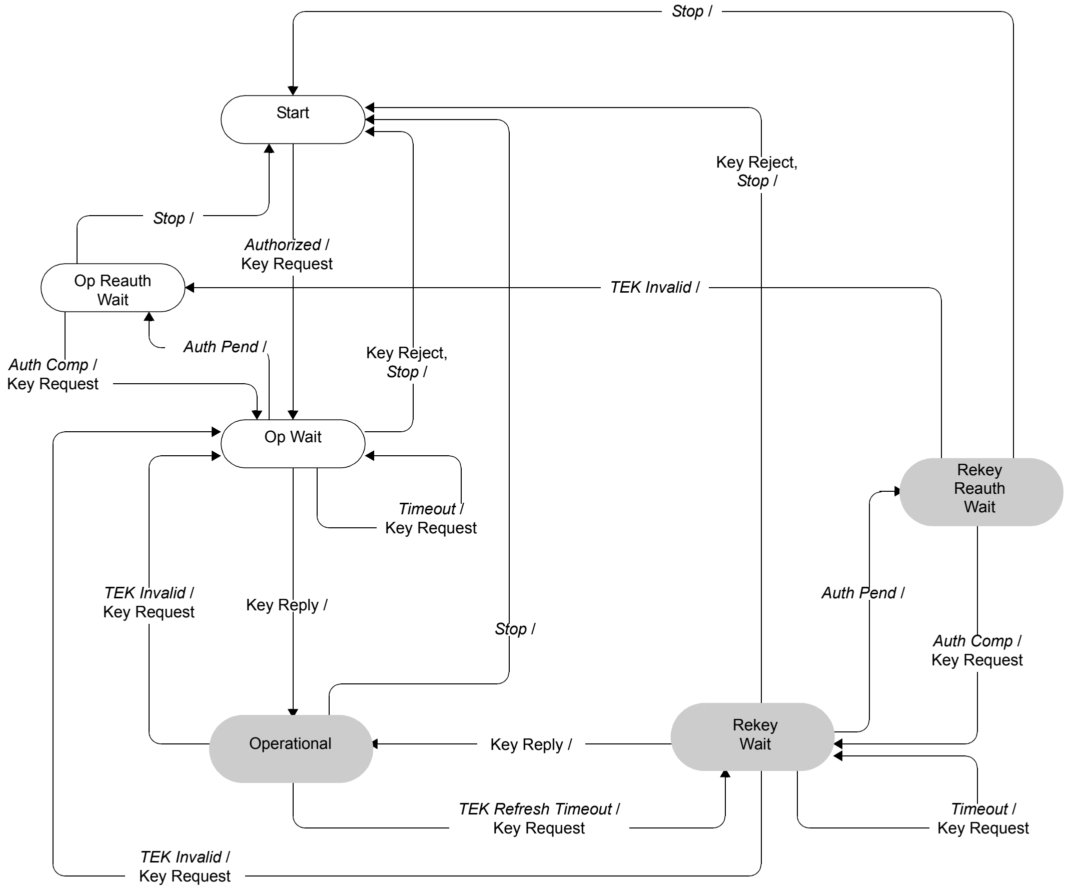


Figure 98—TEK state machine flow diagram

Table 73—TEK FSM state transition matrix

State ***Event or Rcvd Message***	**(A)** **Start**	**(B)** **Op Wait**	**(C)** **Op Reauth Wait**	**(D)** **Op**	**(E)** **Rekey Wait**	**(F)** **Rekey Reauth Wait**
(1) *Stop*		Start	Start	Start	Start	Start
(2) *Authorized*	Op Wait					
(3) *Auth Pend*		Op Reauth Wait			Rekey Reauth Wait	
(4) *Auth Comp*			Op Wait			Rekey Wait
(5) *TEK Invalid*				Op Wait	Op Wait	Op Reauth Wait
(6) *Timeout*		Op Wait			Rekey Wait	
(7) *TEK Refresh Timeout*				Rekey Wait		
(8) Key Reply		Operational			Operational	
(9) Key Reject		Start			Start	

7.2.5.1 States

Start: This is the initial state of the FSM. No resources are assigned to or used by the FSM in this state—e.g., all timers are off, and no processing is scheduled.

Operational Wait (Op Wait): The TEK state machine has sent its initial request (Key Request) for its SAID's keying material (TEK and CBC initialization vector), and is waiting for a reply from the BS.

Operational Reauthorize Wait (Op Reauth Wait): The wait state the TEK state machine is placed in if it does not have valid keying material while the Authorization state machine is in the in the middle of a reauthorization cycle.

Operational: The SS has valid keying material for the associated SAID.

Rekey Wait: The TEK Refresh Timer has expired and the SS has requested a key update for this SAID. Note that the newer of its two TEKs has not expired and can still be used for both encrypting and decrypting data traffic.

Rekey Reauthorize Wait (Rekey Reauth Wait): The wait state the TEK state machine is placed in if the TEK state machine has valid traffic keying material, has an outstanding request for the latest keying material, and the Authorization state machine initiates a reauthorization cycle.

7.2.5.2 Messages

Note that the message formats are defined in detail in 6.2.2.3.9.

Key Request: Request a TEK for this SAID. Sent by the SS to the BS and authenticated with keyed message digest. The message authentication key is derived from the AK.

Key Reply: Response from the BS carrying the two active sets of traffic keying material for this SAID. Sent by the BS to the SS, it includes the SAID's traffic encryption keys, triple DES encrypted with a key encryption key derived from the AK. The Key Reply message is authenticated with a keyed message digest; the authentication key is derived from the AK.

Key Reject: Response from the BS to the SS to indicate this SAID is no longer valid and no key will be sent. The Key Reject message is authenticated with a keyed message digest; the authentication key is derived from the Authorization Key.

TEK Invalid: The BS sends an SS this message if it determines that the SS encrypted an uplink PDU with an invalid TEK, i.e., an SAID's TEK key sequence number, contained within the received PDU's MAC Header, is out of the BS's range of known, valid sequence numbers for that SAID.

7.2.5.3 Events

Stop: Sent by the Authorization FSM to an active (non-START state) TEK FSM to terminate TEK FSM and remove the corresponding SAID's keying material from the SS's key table. See Figure 97.

Authorized: Sent by the Authorization FSM to a non-active (START state) TEK FSM to notify TEK FSM of successful authorization. See Figure 97.

Authorization Pending (Auth Pend): Sent by the Authorization FSM to TEK FSM to place TEK FSM in a wait state while Authorization FSM completes re-authorization. See Figure 97.

Authorization Complete (Auth Comp): Sent by the Authorization FSM to a TEK FSM in the Operational Reauthorize Wait or Rekey Reauthorize Wait states to clear the wait state begun by the prior Authorization Pending event. See Figure 97.

TEK Invalid: This event is triggered by either an SS's data packet decryption logic or by the receipt of a TEK Invalid message from the BS.

An SS's data packet decryption logic triggers a TEK Invalid event if it recognizes a loss of TEK key synchronization between itself and the encrypting BS. For example, an SAID's TEK key sequence number, contained within the received downlink MAC PDU Header, is out of the SS's range of known sequence numbers for that SAID.

A BS sends an SS a TEK Invalid message, triggering a TEK Invalid event within the SS, if the BS's decryption logic recognizes a loss of TEK key synchronization between itself and the SS.

Timeout: A retry timer timeout. Generally, the particular request is retransmitted.

TEK Refresh Timeout: The TEK refresh timer timed out. This timer event signals the TEK state machine to issue a new Key Request in order to refresh its keying material. The refresh timer is set to fire a configurable duration of time (*TEK Grace Time*) before the expiration of the newer TEK the SS currently holds. This is configured via the BS to occur after the scheduled expiration of the older of the two TEKs.

7.2.5.4 Parameters

All configuration parameter values are specified in TFTP downloaded SS Configuration File (see 9.2).

Operational Wait Timeout: Timeout period between sending of Key Request messages from the Op Wait state (see 11.2.19.4).

Rekey Wait Timeout: Timeout period between sending of Key Request messages from the Rekey Wait state (see 11.2.19.5).

TEK Grace Time: Time interval, in seconds, before the estimated expiration of a TEK that the SS starts rekeying for a new TEK. TEK Grace Time is specified in a configuration setting within the TFTP-downloaded SS Configuration File and is the same across all SAIDs (see 11.2.19.6).

7.2.5.5 Actions

Actions taken in association with state transitions are listed by <event> (<rcvd message>) --> <state>:

1-B Op Wait (*Stop*) → Start

a) clear Key Request retry timer
b) terminate TEK FSM

1-C Op Reauth Wait (*Stop*) → Start

a) terminate TEK FSM

1-D Operational (*Stop*) → Start

a) clear TEK refresh timer, which is timer set to go off *“TEK Grace Time”* seconds prior to the TEK’s scheduled expiration time
b) terminate TEK FSM
c) remove SAID keying material from key table

1-E Rekey Wait (*Stop*) → Start

a) clear Key Request retry timer
b) terminate TEK FSM
c) remove SAID keying material from key table

1-F Rekey Reauth Wait (*Stop*) → Start

a) terminate TEK FSM
b) remove SAID keying material from key table

2-A Start (*Authorized*) → Op Wait

a) send Key Request Message to BS
b) set Key Request retry timer to Operational Wait Timeout

3-B Op Wait (*Auth Pend*) → Op Reauth Wait

a) clear Key Request retry timer

3-E Rekey Wait (*Auth Pend*) → Rekey Reauth Wait

a) clear Key Request retry timer

4-C Op Reauth Wait (*Auth Comp*) → Op Wait

a) send Key Request message to BS
b) set Key Request retry timer to Operational Wait Timeout

4-F Rekey Reauth Wait (*Auth Comp*) → Rekey Wait

a) send Key Request message to BS
b) set Key Request retry timer to Rekey Wait Timeout

5-D Operational (*TEK Invalid*) → Op Wait

a) clear TEK refresh timer
b) send Key Request message to BS
c) set Key Request retry timer to Operational Wait Timeout
d) remove SAID keying material from key table

5-E Rekey Wait (*TEK Invalid*) → Op Wait

a) clear Key Request retry timer
b) send Key Request message to BS
c) set Key Request retry timer to Operational Wait Timeout
d) remove SAID keying material from key table

5-F Rekey Reauth Wait (*TEK Invalid*) → Op Reauth Wait

a) remove SAID keying material from key table

6-B Op Wait (*Timeout*) → Op Wait

a) send Key Request message to BS
b) set Key Request retry timer to Operational Wait Timeout

6-E Rekey Wait (*Timeout*) → Rekey Wait

a) send Key Request message to BS
b) set Key Request retry timer to Rekey Wait Timeout

7-D Operational (*TEK Refresh Timeout*) → Rekey Wait

a) send Key Request message to BS
b) set Key Request retry timer to Rekey Wait Timeout

8-B Op Wait (Key Reply) → Operational

a) clear Key Request retry timer
b) process contents of Key Reply message and incorporate new keying material into key database
c) set the TEK refresh timer to go off "TEK Grace Time" seconds prior to the key's scheduled expiration

8-E Rekey Wait (Key Reply) → Operational

a) clear Key Request retry timer
b) process contents of Key Reply message and incorporate new keying material into key database
c) set the TEK refresh timer to go off "TEK Grace Time" seconds prior to the key's scheduled expiration

9-B Op Wait (Key Reject) → Start

a) clear Key Request retry timer
b) terminate TEK FSM

9-E Rekey Wait (Key Reject) → Start

a) clear Key Request retry timer
b) terminate TEK FSM
c) remove SAID keying material from key table

7.3 Dynamic SA creation and mapping

Dynamic Security Associations are SAs that a BS establishes and eliminates dynamically in response to the enabling or disabling of specific downlink service flows. SSs learn the mapping of a particular privacy-enabled service flow to that flow's dynamically assigned SA through the exchange of DSx messages.

7.3.1 Dynamic SA creation

The BS may dynamically establish SAs by issuing an SA Add message. Upon receiving an SA Add message, the SS shall start a TEK state machine for each SA listed in the message.

7.3.2 Dynamic mapping of SA

When creating a new service flow, an SS may request an existing SA be used by passing the SAID of the SA in a DSA-REQ or DSC-REQ message. The BS checks the SS's authorization for the requested SA and generates appropriate response using a DSA-RSP or DSC-RSP message correspondingly.

With BS-initiated dynamic service creations, a BS may also map a new service flow to an existing SA that is supported by a specific SS. The SAID of the SA shall be communicated to the SS in a DSA-REQ or DSC-REQ message.

7.4 Key usage

7.4.1 BS key usage

The BS is responsible for maintaining keying information for all SAs. The PKM protocol defined in this specification describes a mechanism for synchronizing this keying information between a BS and its client SS.

7.4.1.1 AK key lifetime

After an SS completes basic capabilities negotiation, it shall initiate an authorization exchange with its BS. The BS's first receipt of an Auth Request message from the unauthorized SS shall initiate the activation of a new Authorization Key (AK), which the BS sends back to the requesting SS in an Auth Reply message. This AK shall remain active until it expires according to its predefined *AK Lifetime*, a BS system configuration parameter.

The AK's active lifetime a BS reports in an Authorization Reply message shall reflect, as accurately as an implementation permits, the remaining lifetimes of AK at the time the Authorization Reply message is sent.

If an SS fails to reauthorize before the expiration of its current AK, the BS shall hold no active AKs for the SS and shall consider the SS *unauthorized*. A BS shall remove from its keying tables all TEKs associated with an unauthorized SS's Primary SA.

7.4.1.2 AK transition period on BS side

The BS shall always be prepared to send an AK to an SS upon request. The BS shall be able to support two simultaneously active AKs for each client SS. The BS has two active AKs during an Authorization Key transition period; the two active keys have overlapping lifetimes.

An AK transition period begins when the BS receives an Auth Request message from an SS and the BS has a single active AK for that SS. In response to this Auth Request, the BS activates a second AK [see point (a) and (d) in Figure 99], which shall have a key sequence number one greater (modulo 16) than that of the existing AK and shall be sent back to the requesting SS in an Auth Reply message. The BS shall set the active lifetime of this second AK to be the remaining lifetime of the first AK (between points (a) and (c) in Figure 99), plus the predefined *AK Lifetime*; thus, the second, "newer" key shall remain active for one *AK Lifetime* beyond the expiration of the first, "older" key. The key transition period shall end with the expiration of the older key. This is depicted on the right-hand side of Figure 99.

As long as the BS is in the midst of an SS's AK transition period, and thus is holding two active AKs for that SS, it shall respond to Auth Request messages with the newer of the two active keys. Once the older key expires, an Auth Request shall trigger the activation of a new AK, and the start of a new key transition period.

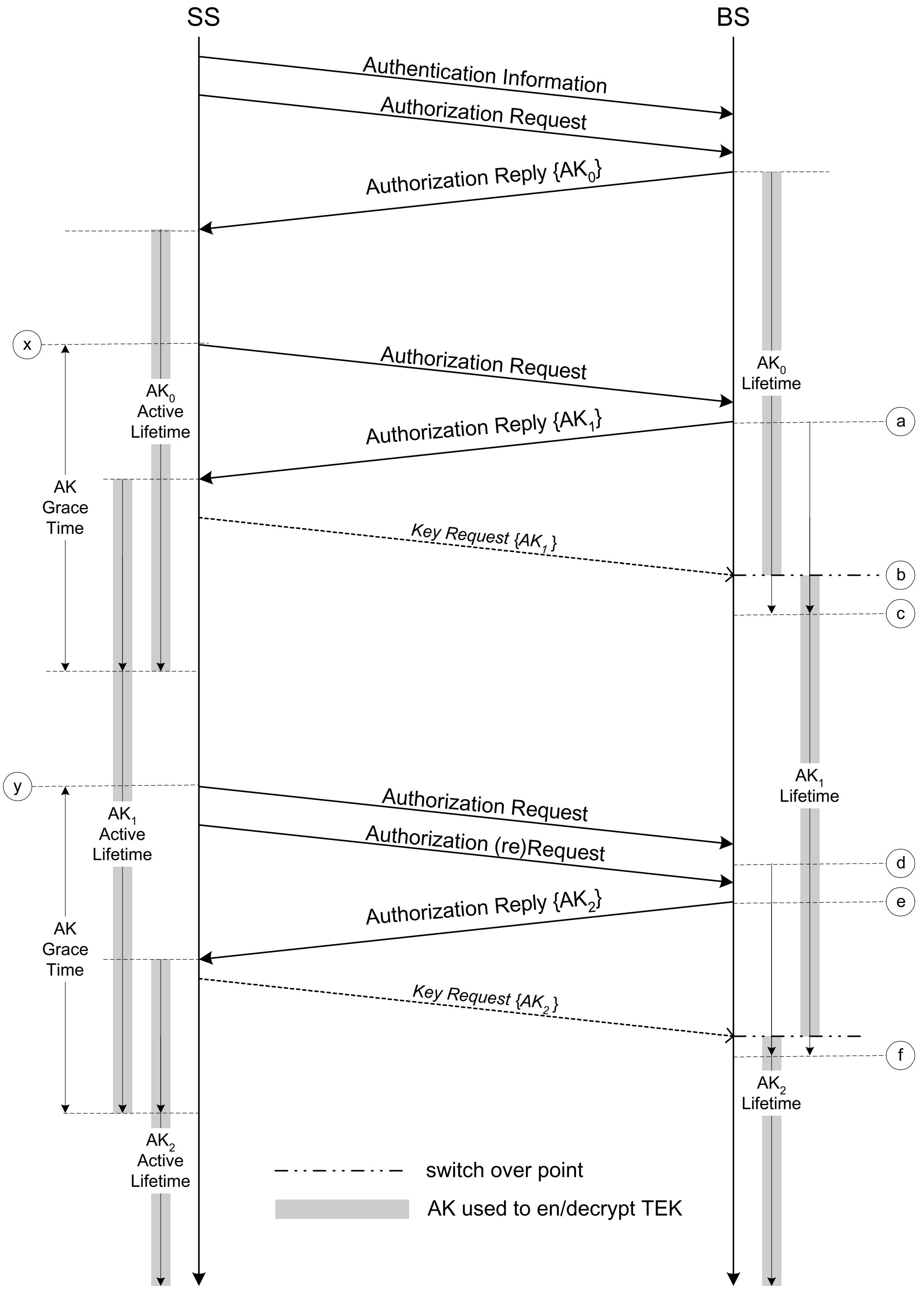


Figure 99—AK management in BS and SS

7.4.1.3 BS usage of AK

The BS shall use keying material derived from the SS's AK for the following:

a) verifying the HMAC-Digests in Key Request messages received from that SS,

b) calculating the HMAC-Digests it writes into Key Reply, Key Reject, and TEK Invalid messages sent to that SS, and

c) encrypting the TEK in the Key Reply messages it sends to that SS.

A BS shall use an HMAC_KEY_U (see 7.5.4.3) derived from one of the SS's active AKs to verify the HMAC-Digest in Key Request messages received from the SS. The AK Key Sequence Number accompanying each Key Request message allows the BS to determine which HMAC_KEY_U was used to authenticate the message. If the AK Key Sequence Number indicates the newer of the two AKs, the BS shall identify this as an *implicit acknowledgment* that the SS has obtained the newer of the SS's two active AKs [see points (b) in Figure 99].

A BS shall use an HMAC_KEY_D derived from the active AK selected above (see 7.5.4.3) when calculating HMAC-Digests in Key Reply, Key Reject, and TEK Invalid message. When sending Key Reply, Key Reject, or TEK Invalid messages within a key transition period (i.e., when two active AKs are available), if the newer key has been implicitly acknowledged, the BS shall use the newer of the two active AKs. If the newer key has not been implicitly acknowledged, the BS shall use the older of the two active AKs to derive the KEK and the HMAC_KEY_D.

The BS shall use a KEK derived from an active AK when encrypting the TEK in the Key Reply messages. The right-hand side of Figure 99 illustrates the BS's policy regarding its use of AKs, where the shaded portion of an AK's lifetime indicates the time period during which that AK shall be used to derive the HMAC_KEY_U, HMAC_KEY_D, and KEK.

For calculating the HMAC digest in the HMAC Tuple attribute, the BS shall use the HMAC_KEY_U and HMAC_KEY_D derived from one of the active AKs. For signing messages, if the newer AK has been implicitly acknowledged, the BS shall use the newer of the two active AKs to derive the HMAC_KEY_D. If the newer key has not been implicitly acknowledged, the BS shall use the older of the two active AKs to derive the HMAC_KEY_D. The HMAC Key Sequence Number in the HMAC Tuple, equal to the AK's sequence number from which the HMAC_KEY_D was derived, enables the SS to correctly determine which HMAC_KEY_D was used for message authentication.

When receiving messages containing the HMAC Tuple attribute, the BS shall use the HMAC_KEY_U indicated by the HMAC Key Sequence Number to authenticate the messages.

7.4.1.4 TEK lifetime

The BS shall maintain two sets of active TEKs (and their associated Initialization Vectors, or IVs) per SAID, corresponding to two successive generations of keying material. The two generations of TEKs shall have overlapping lifetimes determined by *TEK Lifetime*, a predefined BS system configuration parameter. The newer TEK shall have a key sequence number one greater (modulo 4) than that of the older TEK. Each TEK becomes active halfway through the lifetime of its predecessor and expires halfway through the lifetime of its successor. Once a TEK's lifetime expires, the TEK becomes inactive and shall no longer be used.

The Key Reply messages sent by a BS contain TEK parameters for the two active TEKs. The TEKs' active lifetimes a BS reports in a Key Reply message shall reflect, as accurately as an implementation permits, the remaining lifetimes of these TEKs at the time the Key Reply message is sent.

7.4.1.5 BS usage of TEK

The BS transitions between the two active TEKs differently, depending on whether the TEK is used for downlink or uplink traffic. For each of its SAIDs, the BS shall transition between active TEKs according to the following rules:

a) At expiration of the older TEK, the BS shall immediately transition to using the newer TEK for encryption.

b) The uplink transition period begins from the time the BS sends the newer TEK in a Key Reply Message and concludes once the older TEK expires.

It is the responsibility of the SS to update its keys in a timely fashion; the BS shall transition to a new downlink encryption key regardless of whether a client SS has retrieved a copy of that TEK.

The BS uses the two active TEKs differently, depending on whether the TEK is used for downlink or uplink traffic. For each of its SAIDs, the BS shall use the two active TEKs according to the following rules:

a) The BS shall use the older of the two active TEKs for encrypting downlink traffic.

b) The BS shall be able to decrypt uplink traffic using either the older or newer TEK.

Note that the BS encrypts with a given TEK for only the second half of that TEK's total lifetime. The BS is able, however, to decrypt with a TEK for the TEK's entire lifetime.

The right-hand side of Figure 100 illustrates the BS's management of an SA's TEKs, where the shaded portion of a TEK's lifetime indicates the time period during which that TEK shall be used to encrypt MAC PDU payloads.

7.4.2 SS key usage

The SS is responsible for sustaining authorization with its BS and maintaining an active Authorization Key. An SS shall be prepared to use its two most recently obtained AKs according to the following manner.

7.4.2.1 SS reauthorization

AKs have a limited lifetime and shall be periodically refreshed. An SS refreshes its Authorization Key by reissuing an Auth Request to the BS. The Authorization State Machine (7.2.4) manages the scheduling of Auth Requests for refreshing AKs.

An SS's Authorization state machine schedules the beginning of reauthorization a configurable duration of time, the *Authorization Grace Time*, [see points (x) and (y) in Figure 99], before the SS's latest AK is scheduled to expire. The Authorization Grace Time is configured to provide an SS with an authorization retry period that is sufficiently long to allow for system delays and provide adequate time for the SS to successfully complete an Authorization exchange before the expiration of its most current AK.

Note that the BS does not require knowledge of the Authorization Grace Time. The BS, however, shall track the lifetimes of its Authorization Keys and shall deactivate a key once it has expired.

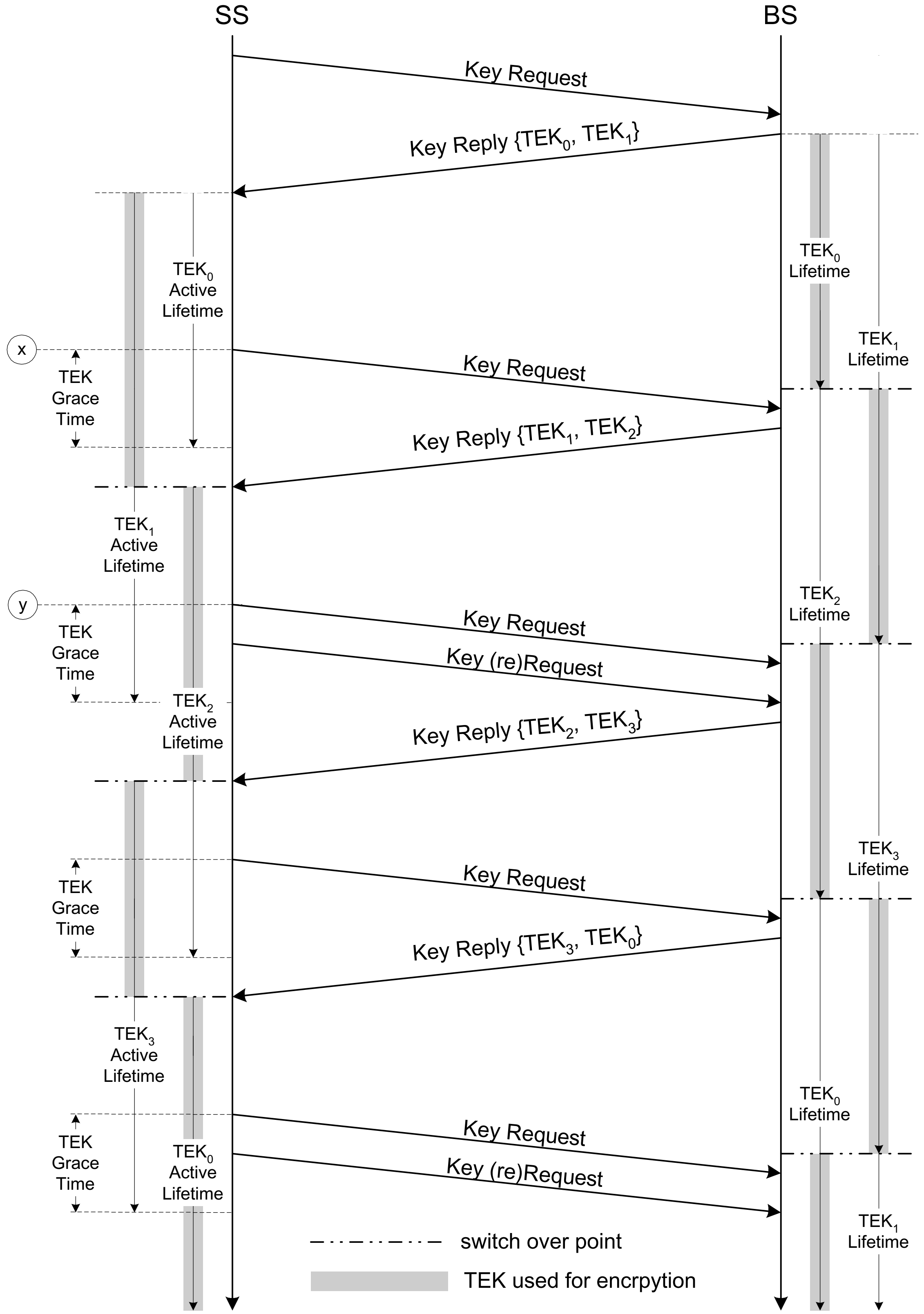


Figure 100—TEK management in BS and SS

7.4.2.2 SS usage of AK

An SS shall use the HMAC_KEY_U derived from the newer of its two most recent AKs when calculating the HMAC-Digests it attaches to Key Request messages.

The SS shall be able to use the HMAC_KEY_D derived from either of its two most recent AKs to authenticate Key Reply, Key Reject, and TEK Reject messages. The SS shall be able to decrypt an encrypted TEK in a Key Reply message with the KEK derived from either of its two most recent AKs. The SS shall use the accompanying AK Key Sequence Number to determine which set of keying material to use.

The left-hand side of Figure 99 illustrates an SS's maintenance and usage of its AKs, where the shaded portion of an AK's lifetime indicates the time period during which that AK shall be used to decrypt TEKs. Even though it is not part of the message exchange, Figure 99 also shows the implicit acknowledgement of the reception of a new AK via the transmission of a Key Request message using the key sequence of the new AK.

An SS shall use the HMAC_KEY_U derived from the newer of its two most recent AKs when calculating the HMAC-Digests of the HMAC Tuple attribute.

7.4.2.3 SS usage of TEK

An SS shall be capable of maintaining two successive sets of traffic keying material per authorized SAID. Through operation of its TEK state machines, an SS shall request a new set of traffic keying material a configurable amount of time, the *TEK Grace Time* [see points (x) and (y) in Figure 100], before the SS's latest TEK is scheduled to expire.

For each of its authorized SAIDs, the SS:

a) shall use the newer of its two TEKs to encrypt uplink traffic, and

b) shall be able to decrypt downlink traffic encrypted with either of the TEKs.

The left-hand side of Figure 100 illustrates the SS's maintenance and usage of an SA's TEKs, where the shaded portion of a TEK's lifetime indicates the time period during which that TEK shall be used to encrypt MAC PDU payloads.

7.5 Cryptographic methods

This subclause specifies the cryptographic algorithms and key sizes used by the PKM protocol. All SS and BS implementations shall support the method of packet data encryption defined in 7.5.1, encryption of the TEK as specified in 7.5.2, and message digest calculation as specified in 7.5.3.

7.5.1 Data encryption with DES

If the Data Encryption Algorithm Identifier in the Cryptographic Suite of an SA equals 0x01, data on connections associated with that SA shall use the Cipher Block Chaining (CBC) mode of the US Data Encryption Standard (DES) algorithm [FIPS 46-3, FIPS 74, FIPS 81] to encrypt the MAC PDU payloads.

The CBC IV shall be calculated as follows: in the downlink, the CBC shall be initialized with the Exclusive-OR (XOR) of (1) the IV parameter included in the TEK keying information, and (2) the content of the PHY Synchronization field of the latest DL-MAP. In the uplink, the CBC shall be initialized with the XOR of (1) the IV parameter included in the TEK keying information, and (2) the content of the PHY Synchronization field of the DL-MAP that is in effect when the UL-MAP for the uplink transmission is created/received.

Residual termination block processing shall be used to encrypt the final block of plaintext when the final block is less than 64 bits. Given a final block having *n* bits, where *n* is less than 64, the next-to-last ciphertext block shall be DES encrypted a second time, using the Electronic Code Book (ECB) mode, and the most significant *n* bits of the result are XORed with the final *n* bits of the payload to generate the short final cipher block. In order for the receiver to decrypt the short final cipher block, the receiver DES encrypts the next-to-last ciphertext block, using the ECB mode, and XORs the most significant *n* bits with the short final cipher block in order to recover the short final cleartext block. This encryption procedure is depicted in Figure 9.4 of Schneier [B18].

In the special case when the payload portion of the MAC PDU is less than 64 bits, the IV shall be DES encrypted and the most significant *n* bits of the resulting ciphertext, corresponding to the number of bits of the payload, shall be XORed with the *n* bits of the payload to generate the short cipher block.[15]

7.5.2 Encryption of TEK with 3-DES

This method of encrypting the TEK shall be used for SAs with the TEK Encryption Algorithm Identifier in the Cryptographic Suite equal to 0x01.

The BS encrypts the value fields of the TEK in the Key Reply messages it sends to client SS. This field is encrypted using two-key triple DES in the encrypt-decrypt-encrypt (EDE) mode [B18]:

encryption: C = Ek1[Dk2[Ek1[P]]]
decryption: P = Dk1[Ek2[Dk1[C]]]
P = Plaintext 64-bit TEK
C = Ciphertext 64-bit TEK
k1 = left-most 64 bits of the 128-bit KEK
k2 = right-most 64 bits of the 128-bit KEK
E[ ] = 56-bit DES ECB (electronic code book) mode encryption
D[ ] = 56-bit DES ECB decryption

Subclause 7.5.4 below describes how the KEK is derived from the AK.

7.5.3 Calculation of HMAC digests

The calculation of the keyed hash in the HMAC-Digest attribute and the HMAC Tuple shall use the HMAC (IETF RFC 2104) with the SHA-1 hash algorithm (FIPS 180-1). The downlink authentication key HMAC_KEY_D shall be used for authenticating messages in the downlink direction. The uplink authentication key HMAC_KEY_U shall be used for authenticating messages in the uplink direction. Uplink and downlink message authentication keys are derived from the AK (see 7.5.4 below for details). The HMAC Sequence number in the HMAC Tuple shall be equal to the AK Sequence Number of the AK from which the HMAC_KEY_x was derived.

The digest shall be calculated over the entire MAC Management Message with the exception of the HMAC-Digest and HMAC Tuple attributes.

7.5.4 Derivation of TEKs, KEKs, and message authentication keys

The BS generates AKs, TEKs and IVs. A random or pseudo-random number generator shall be used to generate AKs and TEKs. A random or pseudorandom number generator may also be used to generate IVs.

[15] If two or more PDUs with less than 8-byte payloads are transmitted in the same frame using the same SA, the XOR of the payload plaintexts can be found easily. In practice, this situation is very unlikely to occur, as payloads are typically larger than 8 bytes. In the case that multiple payloads of less than 8 bytes are to be transmitted in the same frame on the same SA and service, packing of the short SDUs into a single PDU will eliminate this weakness. If the SDUs are for different services, packing the SDUs with zero-length fictitious SDUs allows the use of the packing subheader to extend the size of the PDU to at least 8 bytes.

Regardless of how they are generated, IVs shall be unpredictable. Recommended practices for generating random numbers for use within cryptographic systems are provided in IETF RFC 1750 [B10].

7.5.4.1 DES Keys

FIPS 81 defines 56-bit DES keys as 8-byte (64-bit) quantities where the seven most significant bits (i.e., seven left-most bits) of each byte are the independent bits of a DES key, and the least significant bit (i.e., right-most bit) of each byte is a parity bit computed on the preceding seven independent bits and adjusted so that the byte has odd parity.

PKM does not require odd parity. The PKM protocol generates and distributes 8-byte DES keys of arbitrary parity, and it requires that implementations ignore the value of the least significant bit of each.

7.5.4.2 3-DES KEKs

The keying material for two-key triple DES consists of two distinct (single) DES keys.

The 3-DES KEK used to encrypt the TEK is derived from a common AK. The KEK shall be derived as follows:

KEK=Truncate(SHA(K_PAD_KEK | AK),128)
K_PAD_KEK=0x53 repeated 64 times, i.e., a 512 bit string.

Truncate(x,*n*) denotes the result of truncating x to its left-most *n* bits.

SHA(*x*|*y*) denotes the result of applying the SHA-1 function to the concatenated bit strings *x* and *y*.

The keying material of 3-DES consists of two distinct DES keys. The 64 most significant bits of the KEK shall be used in the encrypt operation. The 64 least significant bits shall be used in the decrypt operation.

7.5.4.3 HMAC authentication keys

The HMAC authentication keys are derived as follows:

HMAC_KEY_D=SHA(H_PAD_D|AK)
HMAC_KEY_U=SHA(H_PAD_U|AK).

with

H_PAD_D=0x3A repeated 64 times
H_PAD_U=0x5C repeated 64 times.

7.5.5 Public-key encryption of authorization key

AKs in Auth Reply messages shall be RSA public-key encrypted, using the SS's public key. The protocol uses 65537 (0x010001) as its public exponent and a modulus length of 1024 bits. The PKM protocol employs the RSAES-OAEP encryption scheme (PKCS #1). RSAES-OAEP requires the selection of a hash function, a mask-generation function, and an encoding parameter string. The default selections specified in PKCS #1 shall be used when encrypting the AK. These default selections are SHA-1 for the hash function, MGF1 with SHA-1 for the mask-generation function, and the empty string for the encoding parameter string.

7.5.6 Digital signatures

The Protocol employs the RSA Signature Algorithm [PKCS #1] with SHA-1 [FIPS 186-2] for both of its certificate types.

As with its RSA encryption keys, Privacy uses 65537 (0x010001) as the public exponent for its signing operation. Manufacturer CAs shall employ signature key modulus lengths of at least 1024 bits and no greater than 2048 bits.

7.6 Certificate profile

7.6.1 Certificate format

This subclause describes the X.509 [IETF RFC 2459] Version 3 certificate format and certificate extensions used in IEEE 802.16-2001 compliant SSs. Table 74 below summarizes the basic fields of an X.509 Version 3 certificate.

Table 74—Basic fields of an X.509 Version 3 certificate

X.509 v3 field	Description
tbsCertificate.version	Indicates the X.509 certificate version. Always set to v3 (value of 2)
tbsCertificate.serialNumber	Unique integer the issuing CA assigns to the certificate.
tbsCertificate.signature	Object Identifier (OID) and optional parameters defining algorithm used to sign the certificate. This field shall contain the same algorithm identifier as the signatureAlgorithm field below.
tbsCertificate.issuer	Distinguished Name of the CA that issued the certificate.
tbsCertificate.validity	Specifies when the certificate becomes active and when it expires.
tbsCertificate.subject	Distinguished Name identifying the entity whose public key is certified in the subjectpublic key information field.
tbsCertificate.subjectPublicKeyInfo	Field contains the public key material (public key and parameters) and the identifier of the algorithm with which the key is used.
tbsCertificate.issuerUniqueID	Optional field to allow reuse of issuer names over time.
tbsCertificate.subjectUnique ID	Optional field to allow reuse of subject names over time.
tbsCertificate.extensions	The extension data.
signatureAlgorithm	OID and optional parameters defining algorithm used to sign the certificate. This field shall contain the same algorithm identifier as the signature field in tbsCertificate.
signatureValue	Digital signature computed upon the ASN.1 DER encoded tbsCertificate.

All certificates described in this specification shall be signed with the RSA signature algorithm using SHA-1 as the one-way hash function. The RSA signature algorithm is described in PKCS #1; SHA-1 is described in FIPS 180-1. Restrictions posed on the certificate values are described below:

7.6.1.1 tbsCertificate.validity.notBefore and tbsCertificate.validity.notAfter

SS certificates shall not be renewable and shall thus have a validity period greater than the operational lifetime of the SS. A Manufacturer CA certificate's validity period should exceed that of the SS certificates it issues. The validity period of an SS certificate shall begin with the date of generation of the device's certificate; the validity period should extend out to at least 10 years after that manufacturing date. Validity periods shall be encoded as UTCTime. UTCTime values shall be expressed Greenwich Mean Time (Zulu) and shall include seconds (i.e., times are YYMMDDHHMMSSZ), even where the number of seconds is zero.

7.6.1.2 tbsCertificate.serialNumber

Serial numbers for SS certificates signed by a particular issuer shall be assigned by the manufacturer in increasing order. Thus, if the tbsCertificate.validity.notBefore field of one certificate is greater than the tbsCertificate.validity.notBefore field of another certificate, then the serial number of the first certificate shall be greater than the serial number of the second certificate.

7.6.1.3 tbsCertificate.signature and signatureAlgorithm

All certificates described in this specification shall be signed with the RSA signature algorithm, using SHA-1 as the one-way hash function. The RSA signature algorithm is described in PKCS #1; SHA-1 is described in FIPS 180-1. The ASN.1 Object Identifier (OID) used to identify the "SHA-1 with RSA" signature algorithm is

```
sha-1WithRSAEncryption OBJECT IDENTIFIER ::=
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 5}
```

When the sha-1WithRSAEncryption OID appears within the ASN.1 type AlgorithmIdentifier, as is the case with both tbsCertificate.signature and signatureAlgorithm, the parameters component of that type is the ASN.1 type NULL.

7.6.1.4 tbsCertificate.issuer and tbsCertificate.subject

X.509 Names are SEQUENCES of RelativeDistinguishedNames, which are in turn SETs of AttributeTypeAndValue. AttributeTypeAndValue is a SEQUENCE of an AttributeType (an OBJECT IDENTIFIER) and an AttributeValue. The value of the countryName attribute shall be a 2-character PrintableString, chosen from ISO 3166; all other AttributeValues shall be encoded as either T.61/TeletexString or PrintableString character strings. The PrintableString encoding shall be used if the character string contains only characters from the PrintableString set. Specifically:

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
'()+,-./:=? and space

The T.61/TeletexString shall be used if the character string contains other characters. The following OIDs are needed for defining issuer and subject Names in PKM certificates:

```
id-at OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 4}
id-at-commonName OBJECT IDENTIFIER ::= {id-at 3}
id-at-countryName OBJECT IDENTIFIER ::= {id-at 6}
id-at-localityName OBJECT IDENTIFIER ::= {id-at 7}
id-at-stateOrProvinceName OBJECT IDENTIFIER ::= {id-at 8}
id-at-organizationName OBJECT IDENTIFIER ::= {id-at 10}
id-at-organizationalUnitName OBJECT IDENTIFIER ::= {id-at 11}
```

The following subclauses describe the attributes which comprise the subject Name forms for each type of PKM certificate. Note that the issuer name form is the same as the subject of the issuing certificate. Additional attribute values that are present but unspecified in the following forms should not cause a device to reject the certificate.

7.6.1.4.1 Manufacturer certificate

```
countryName=<Country of Manufacturer>
[stateOrProvinceName=<state/privince>]
[localityName=<City>]
organizationName=<Company Name>
organizationalUnitName=WirelessMAN
[organizationalUnitName=<Manufacturing Location>]
commonName=<Company Name> <Certification Authority>
```

The countryName, organizationName, and commonName attributes shall be included and shall have the values shown. The organizationalUnitName having the value “WirelessMAN” shall be included. The organizationalUnitName representing manufacturing location should be included. If included, it shall be preceded by the organizationalUnitName having value “WirelessMAN.” The stateOrProvinceName and localityName may be included. Other attributes are not allowed and shall not be included.

7.6.1.4.2 SS certificate

```
countryName=<Country of Manufacturer>
organizationName=<Company Name>
organizationalUnitName=<manufacturing location>
commonName=<Serial Number>
commonName=<MAC Address>
```

The MAC address shall be the SS’s MAC address. It is expressed as six pairs of hexadecimal digits separated by colons (:), e.g., “00:60:21:A5:0A:23.” The Alpha HEX characters (A-F) shall be expressed as uppercase letters.

The organizationalUnitName in an SS certificate, which describes the modem’s manufacturing location, should be the same as the organizationalUnitName in the issuer Name describing a manufacturing location. The countryName, organizationName, organizationalUnitName, and commonName attributes shall be included. Other attributes are not allowed and shall not be included.

7.6.1.5 tbsCertificate.subjectPublicKeyInfo

The tbsCertificate.subjectPublicKeyInfo field contains the public key and the public key algorithm identifier. The tbsCertificate.subjectPublicKeyInfo.algorithm field is an AlgorithmIdentifier structure. The AlgorithemIdentifier's algorithm shall be RSA encryption, identified by the following OID:

```
pkcs-1 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
rsadsi(113549) pkcs(1) 1}
rsaEncryption OBJECT IDENTIFIER ::= { pkcs-1 1}
```

The AlgorithmIdentifier's parameters field shall have ASN.1 type NULL. The RSA public key shall be encoded using the ASN.1 type RSAPublicKey:

```
RSAPublicKey ::= SEQUENCE {
modulus INTEGER, -- n
publicExponent INTEGER, -- e -- }
```

where modulus is the modulus n, and publicExponent is the public exponent e. The DER encoded RSAPublicKey is the value of the BIT STRING tbsCertificate.subjectPublicKeyInfo.subjectPublicKey.

7.6.1.6 tbsCertificate.issuerUniqueID and tbsCertificate.subjectUniqueID

The issuerUniqueID and subjectUniqueID fields shall be omitted for both of the PKM's certificate types.

7.6.1.7 tbsCertificate.extensions

7.6.1.7.1 SS certificates

SS certificates may contain noncritical extensions; they shall not contain critical extensions. If the KeyUsage extension is present, the keyAgreement and keyEncipherment bits shall be turned on, keyCertSign and cRLSign bits shall be turned off, and all other bits should be turned off.

7.6.1.7.2 Manufacturer certificates

Manufacturer certificates may contain the Basic Constraints extension. If included, the Basic Constraints extension may appear as a critical extension or as a noncritical extension. Manufacturer certificates may contain noncritical extensions; they shall not contain critical extensions other than, possibly, the Basic Constraints extension. If the KeyUsage extension is present in a Manufacturer certificate, the keyCertSign bit shall be turned on and all other bits should be turned off.

7.6.1.8 signatureValue

In all three PKM certificate types, the signatureValue contains the RSA (with SHA-1) signature computed over the ASN.1 DER encoded tbsCertificate. The ASN.1 DER encoded tbsCertificate is used as input to the RSA signature function. The resulting signature value is ASN.1 encoded as a bit string and included in the Certificate's signatureValue field.

7.6.2 SS certificate storage and management in the SS

Manufacturer-issued SS certificates shall be stored in SS permanent, write-once memory. SSs that have factory-installed RSA private/public key pairs shall also have factory-installed SS certificates. SSs that rely on internal algorithms to generate an RSA key pair shall support a mechanism for installing a manufacturer-issued SS certificate following key generation.The CA certificate of the Manufacturer CA that signed the SS certificate shall be embedded into the SS software. If a manufacturer issues SS certificates with multiple

Manufacturer CA certificates, the SS software shall include ALL of that manufacturer's CA certificates. The specific Manufacturer CA certificate installed by the SS (i.e., advertised in Authentication Information messages and returned by the MIB object) shall be that identifying the issuer of that modem's SS certificate.

7.6.3 Certificate processing and management in the BS

PKM employs digital certificates to allow BSs to verify the binding between an SS's identity (encoded in an X.509 digital certificate's subject names) and its public key. The BS does this by validating the SS certificate's certification path or chain. Validating the chain means verifying the Manufacturer CA Certificate through some means.

8. Physical layer

8.1 Physical layer (PHY) service specification

8.1.1 Scope

This subclause contains a generic definition of the PHY service specification applicable to all PHY options, each of which may consist of two protocol sublayers as follows:

a) A transmission CS.

b) A Physical Medium Dependent (PMD) sublayer. Each PMD sublayer may require the definition of a unique transmission CS. If the PMD sublayer already provides the defined PHY services, the transmission convergence function might be null.

A PHY may be accompanied by a physical layer management entity. The exact functions of such an entity are not specified in this standard, but in general they are responsible for such functions as gathering of layer-dependent status from the various layer management entities setting the value of layer-specific parameters. The physical layer management entity would typically perform such functions on behalf of general system management entities and would implement standard management protocols.

8.1.2 PHY functions

The protocol reference model for the IEEE Std 802.16-2001 architecture is shown in Figure 1. The PHY service is provided to the MAC entity at both the BS and SS through the PHY service access point (SAP), as shown in Figure 1.

PHY service is described using a set of primitives. The primitives associated with communication between the MAC and the PHY fall into three basic categories:

a) Service primitives that support the data transfer, thus participating as intermediate signals in MAC peer-to-peer interactions. These are the PHY_MACPDU primitives.

b) Service primitives that have local significance and support sublayer-to-sublayer interactions related to layer control. These include the PHY_TXSTART primitives.

c) Service primitives that support management functions, such as the PHY_DCD primitives.

Primitives with names of the form "PHY_*.request" are generated by the MAC and addressed to the PHY to invoke some PHY function(s).

Primitives with names of the form "PHY_*.confirmation" are generated by the PHY and addressed to the MAC to acknowledge a previously received "PHY_*.request" primitive.

Primitives with names of the form "PHY_*.indication" are generated by the PHY and addressed to the MAC as a result of some PHY event.

8.1.3 PHY SAP management

This topic covers PHY SAP service primitives related to physical layer management. Physical layer management functions include frequency adjustment, power management, propagation delay compensation, etc.

8.1.3.1 PHY MIB and generic SET/GET primitives

The management information specific to PHY is represented as a management information base (MIB) for this layer. The physical layer management entity is viewed as "containing" the MIB for this layer. The generic model of MIB-related management is to allow the management system (that is out of scope of this standard) to either GET the value of a MIB attribute, or to SET the value of a MIB attribute. The setting act may require that the layer entity perform certain defined actions.

The same way, PHY SAP may have a set of primitives of the following types:

PHY_GET.request (MIBattribute)
Requests the value of the given (PHY) MIBattribute.
PHY_GET.confirm (status, MIBattribute, MIBattributevalue)
Returns the appropriate MIB attribute value if status = "success," otherwise returns an error indication in the Status field. Possible error status values include "invalid MIB attribute" and "attempt to get write-only MIB attribute."
PHY_SET.request (MIBattribute, MIBattributevalue)
Requests that the indicated MIB attribute be set to the given value. If this MIBattribute implies a specific action, then this requests that the action be performed.
PHY_SET.confirm (status, MIBattribute)
If status = "success," this confirms that the indicated MIB attribute was set to the requested value, otherwise it returns an error condition in status field. If this MIBattribute implies a specific action, then this confirms that the action was performed. Possible error status values include "invalid MIB attribute" and "attempt to set read-only MIB attribute."

8.1.3.2 Parameter sets (vectors)

Several service primitives use parameter vectors. Such a vector is a list of values that may be certain MIB parameters or may be derived by the PHY from MIB parameters or from measurement of the airlink characteristics. The list itself and the set of possible parameter values may vary depending on the PHY details.

A vector may be transferred between the PHY and the MAC as a parameter of a certain primitive. PHY SAP service primitives use the following vectors:

— SCHED_PARAM_VECTOR—the set of PHY parameters that are related to scheduling, such as symbol duration

— DCD_PARAM_VECTOR—the set of PHY parameters related to downlink channel configuration, such as central frequency and active set of burst profiles

— UCD_PARAM_VECTOR—the set of PHY parameters related to uplink channel configuration, such as central frequency and active set of burst profiles

— RNG_REQ_VECTOR—the set of PHY parameters reflecting the transmission adjustments requested from the PHY transmitter

— RNG_IND_VECTOR— the set of PHY parameters reflecting the reception quality information supplied by the PHY receiver

— TXVECTOR—the set of PHY parameters related to the transmit of the PHY PDU, including Tx Power, preamble's specification, burst profiles for each involved burst, amount of data to be transmitted at each burst, etc.

— TXSTATUS—the set of PHY indicators that figure the result of the whole PHY activity triggered by the transmission request from MAC. This may include indication of the overrun/underrun events encountered during the data transfer through the PHY SAP

— RXVECTOR—the set of PHY parameters related to the receipt of the expected PHY PDU, including preamble's specification, burst profiles for each involved burst, amount of data expected to arrive in each burst, etc.

— RXSTATUS—the set of PHY indicators that figure the result of the whole PHY activity triggered by the PHY_RXSTART.request from MAC. This may include indication of the received signal strength, errors encountered by forward error correction (FEC) function, overrun/underrun events encountered during the data transfer through the PHY SAP, etc.

8.1.3.3 PHY_SCHEDPARAM.request

8.1.3.3.1 Function

This primitive requests the set of PHY parameters, such as symbol duration, that are related to scheduling.

8.1.3.3.2 Semantics of the service primitive

```
PHY_SCHEDPARAM.request
        (
        SCHED_PARAM_VECTOR
        )
```

8.1.3.3.3 When generated

Typically, this primitive is generated by the MAC each time as the set of DIUC/UIUCs and/or burst profiles associated with them have changed, e.g., on arrival and handling of the DCD/UCD message.

8.1.3.3.4 Effect of receipt

The PHY entity shall generate a PHY_SCHEDPARAM.response primitive that conveys the requested set of parameters.

8.1.3.4 PHY_SCHEDPARAM.response

8.1.3.4.1 Function

This primitive is to supply to MAC the set of PHY parameters requested by the PHY_SCHEDPARAM.request.

8.1.3.4.2 Semantics of the service primitive

```
PHY_SCHEDPARAM.response
        (
        SCHED_PARAM_VECTOR
        )
```

8.1.3.4.3 When generated

This primitive is generated by the PHY immediately after reception of PHY_SCHEDPARAM.request primitive.

8.1.3.4.4 Effect of receipt

The MAC entity shall get and store the set of values from SCHED_PARAM_VECTOR.

8.1.3.5 PHY_DCD.indication

8.1.3.5.1 Function

This primitive is used by BS PHY to indicate to the local MAC that a signal was received from the physical layer management entity that requests a change in the active set of downlink burst profiles.

8.1.3.5.2 Semantics of the service primitive

```
PHY_DCD.indication
        (
        DCD_PARAM_VECTOR
        )
```

8.1.3.5.3 When generated

This primitive is generated after a signal is received from the Network Management System requesting a change in the active set of burst profiles.

8.1.3.5.4 Effect of receipt

The MAC entity may decide on the change of DIUC set and/or burst profiles associated with them that will result in sending the DCD message(s) to update SSs.

8.1.3.6 PHY_DCD.request

8.1.3.7 Function

This primitive is to supply to SS PHY the set of new values for parameters to be changed according to the previously received DCD message.

8.1.3.7.1 Semantics of the service primitive

```
PHY_DCD.request
        (
        DCD_PARAM_VECTOR
        )
```

8.1.3.7.2 When generated

This primitive is generated by the MAC after reception of the DCD message.

8.1.3.7.3 Effect of receipt

SS PHY performs the requested changes in the active set of burst profiles.

8.1.3.8 PHY_DCD.confirmation

8.1.3.8.1 Function

This primitive is to confirm that the SS PHY completed activity triggered by the PHY_DCD.request

8.1.3.8.2 Semantics of the service primitive

```
PHY_DCD.confirmation
        (
        )
```

8.1.3.8.3 When generated

This primitive is generated by the PHY after reception of the PHY_DCD.request.

8.1.3.8.4 Effect of receipt

None.

8.1.3.9 PHY_DCD primitives example

Figure 101 illustrates a possible scenario for invocation of PHY_DCD primitives.

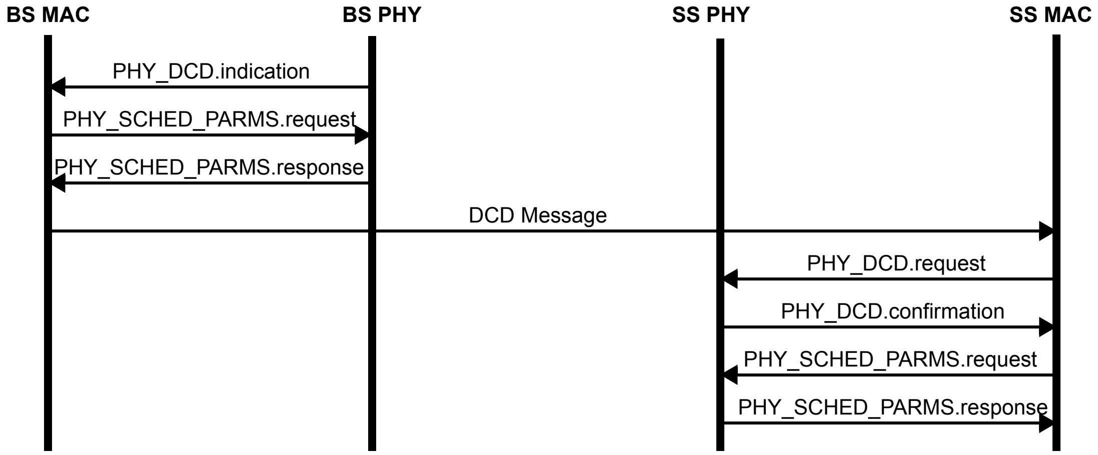


Figure 101—Example of usage of PHY_DCD primitives

8.1.3.10 PHY_UCD.indication

8.1.3.10.1 Function

This primitive is used by BS PHY to indicate to the local MAC that a signal was received from the Network Management System requesting a change in the active set of uplink burst profiles.

8.1.3.10.2 Semantics of the service primitive

```
PHY_UCD.indication
        (
        UCD_PARAM_VECTOR
        )
```

8.1.3.10.3 When generated

This primitive is generated after a signal is received from the physical layer management entity that requests a change in the active set of burst profiles.

8.1.3.10.4 Effect of receipt

MAC may decide to initiate the transmission of UCD message.

8.1.3.11 PHY_UCD.request

8.1.3.11.1 Function

This primitive is to supply to SS PHY the set of new values for parameters to be changed according to the previously received UCD message.

8.1.3.11.2 Semantics of the service primitive

```
PHY_UCD.request
        (
        UCD_PARAM_VECTOR
        )
```

8.1.3.11.3 When generated

This primitive is generated by the MAC after reception of the UCD message.

8.1.3.11.4 Effect of receipt

SS PHY performs the requested changes in the active set of burst profiles.

8.1.3.12 PHY_UCD.confirmation

8.1.3.12.1 Function

This primitive is to confirm that the SS PHY completed activity triggered by the PHY_UCD.request.

8.1.3.12.2 Semantics of the service primitive

```
PHY_UCD.confirmation
        (
        )
```

8.1.3.12.3 When generated

This primitive is generated by the PHY after reception of the PHY_UCD.request.

8.1.3.12.4 Effect of receipt

None.

8.1.3.13 UCD primitives example

Figure 102 illustrates a possible scenario for invocation of PHY_UCD primitives.

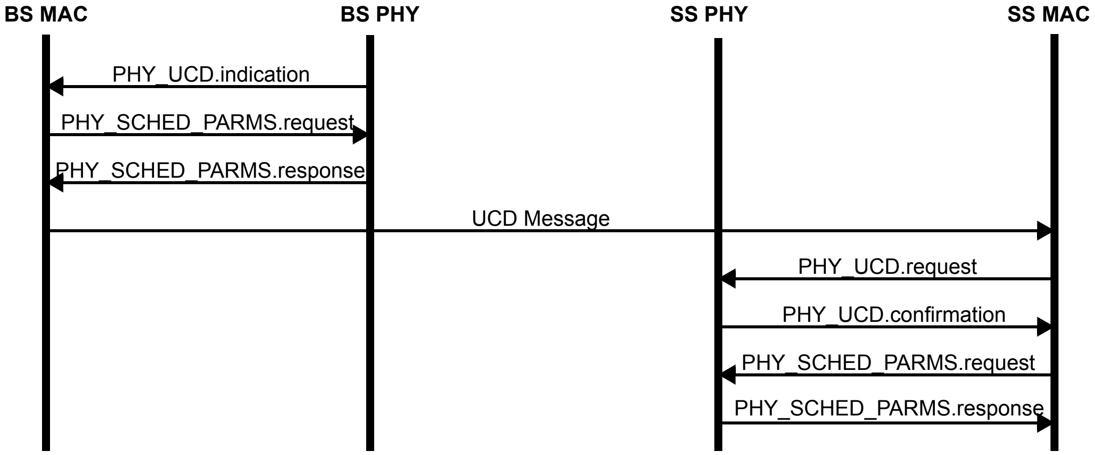


Figure 102—Example of usage of PHY_UCD primitives

8.1.3.14 PHY_RNG.request

8.1.3.14.1 Function

At SS, this primitive is used to transfer to the PHY the parameters received in the RNG-RSP message.

At BS, this primitive is used to transfer to the PHY the parameters received in the RNG-REQ message.

8.1.3.14.2 Semantics of the service primitive

```
PHY_RNG.request
        (
        RNG_REQ_VECTOR
        )
```

8.1.3.14.3 When generated

At SS, this primitive is generated after reception of RNG-RSP message.

At BS, this primitive is generated after reception of RNG-REQ message.

8.1.3.14.4 Effect of receipt

At SS, this primitive causes a transfer of information from the RNG-RSP message to the PHY.

At BS, this primitive causes a transfer of information from the RNG-REQ message to the PHY.

8.1.3.15 PHY_RNG.confirmation

8.1.3.15.1 Function

This primitive is used to confirm the completion of all the needed actions requested by PHY_RNG.request.

8.1.3.15.2 Semantics of the service primitive

```
PHY_RNG.confirmation
        (
        )
```

8.1.3.15.3 When generated

This primitive is generated after completion of all the actions requested by PHY_RNG.request.

8.1.3.15.4 Effect of receipt

None.

8.1.3.16 PHY_RNG.indication

8.1.3.16.1 Function

This primitive appears as a reaction of the PHY to an event that may require a change in PHY parameters.

8.1.3.16.2 Semantics of the service primitive

```
PHY_RNG.indication
        (
        RNG_IND_VECTOR
        )
```

8.1.3.16.3 When generated

This primitive appears when the PHY encounters an event suggesting that PHY parameters could be changed to improve the airlink, such as when a BS encounters a low signal quality from an SS or when an SS can change to a less robust downlink burst profile.

8.1.3.16.4 Effect of receipt

At BS, it is generating and sending unsolicited RNG-RSP message.

At SS, it is generating and sending a RNG-REQ or a DBPC-REQ message.

8.1.3.16.5 Ranging primitives examples

Figure 103 and Figure 104 illustrate possible scenarios of PHY_RNG primitive usage.

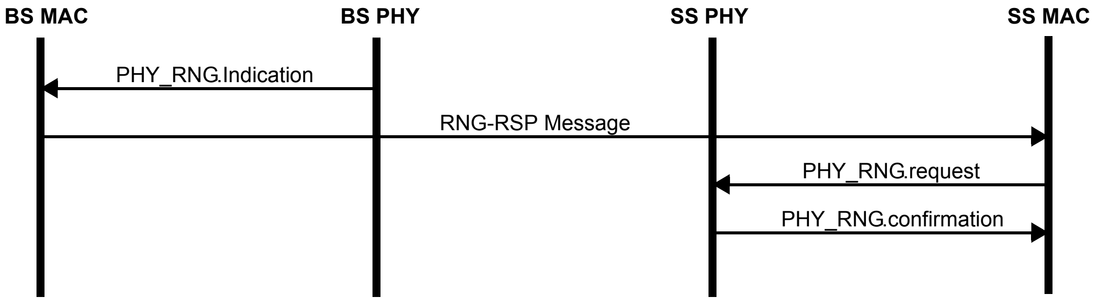


Figure 103—Example of usage of the PHY_RNG primitives with unsolicited RNG-RSP message

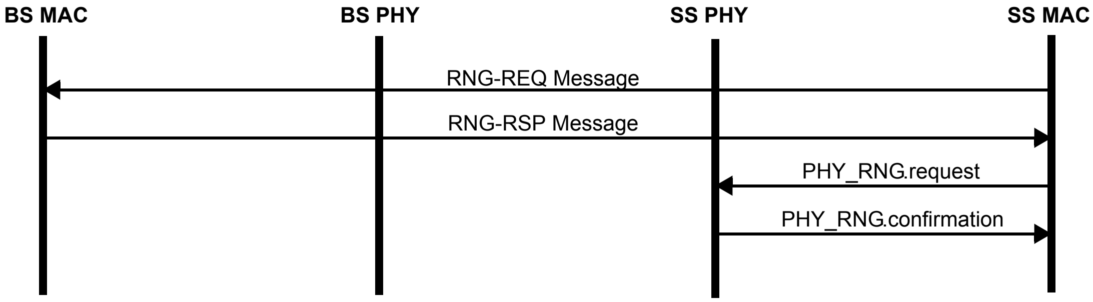


Figure 104—Example of usage of the PHY_RNG primitives: RNG-REQ and RNG-RSP messages

8.1.3.17 PHY_TXSTART.request

8.1.3.17.1 Function

This primitive requests the PHY to start the PHY PDU transmission and carries all the control information needed for the corresponding PHY, such as start and end times of transmission and PHY parameters for multiple bursts, types of preambles, etc.

8.1.3.17.2 Semantics of the service primitive

```
PHY_TXSTART.request
        (
        TXVECTOR
        )
```

8.1.3.17.3 When generated

This primitive is generated by the MAC to initiate PHY PDU transmission.

8.1.3.17.4 Effect of receipt

As a result of reception of this primitive, the PHY sends a confirmation primitive, and at the proper moment, starts transmission of the PHY PDU or continues the transmission of the current PHY PDU with the correspondent change of PHY burst profile.

8.1.3.18 PHY_TXSTART.confirmation

8.1.3.18.1 Function

This primitive confirms reception of PHY_TXSTART.request primitive.

8.1.3.18.2 Semantics of the service primitive

```
PHY_TXSTART.confirmation
        (
        )
```

8.1.3.18.3 When generated

This primitive is generated after reception of PHY_TXSTART.request primitive.

8.1.3.18.4 Effect of receipt

None.

8.1.3.19 PHY_TXSTART.indication

8.1.3.19.1 Function

This primitive indicates to the MAC the actual start of transmission.

8.1.3.19.2 Semantics of the service primitive

```
PHY_TXSTART.indication
        (
        )
```

8.1.3.19.3 When generated

It is generated after the actual start of transmission.

8.1.3.19.4 Effect of receipt

None.

8.1.3.20 PHY_MACPDU.request

8.1.3.20.1 Function

This primitive transfers the MAC message data (DATA parameter) to the PHY and requests the transmission of this data block within the current PHY PDU.

8.1.3.20.2 Semantics of the service primitive

```
PHY_MACPDU.request
          (
          DATA
          )
```

8.1.3.20.3 When generated

This primitive is generated after the corresponding PHY_TXSTART primitive as the data of the next MAC message becomes available.

8.1.3.20.4 Effect of receipt

PHY starts transmission of the MAC Message data.

8.1.3.21 PHY_MACPDU.confirmation

8.1.3.21.1 Function

This primitive confirms that the transmission of the MAC message requested by the PHY_MACPDU primitive has been completed. The TXSTATUS parameter has two possible values: “success” and “failure.” The “failure” may appear, for example, in the case when MAC did not supply the data in proper moment (and PHY transmits padding octets) and, as a consequence, there is not enough place in the PHY PDU for all the MAC messages initially scheduled for transmission.

8.1.3.21.2 Semantics of the service primitive

```
PHY_MACPDU.confirmation
          (
          TXSTATUS
          )
```

8.1.3.21.3 When generated

This primitive is generated after the completion of the transmission of the MAC message requested by the previous PHY_MACPDU primitive.

8.1.3.21.4 Effect of receipt

None.

8.1.3.22 PHY_TXEND.indication

8.1.3.22.1 Function

This primitive indicates the end of the transmission triggered by the PHY_TXSTART primitive.

8.1.3.22.2 Semantics of the service primitive

```
PHY_TXEND.indication
          (
          )
```

8.1.3.22.3 When generated

The primitive is generated at the end of the transmission triggered by the PHY_TXSTART primitive.

8.1.3.22.4 Effect of receipt

None.

8.1.3.23 PHY_RXSTART.request

8.1.3.23.1 Function

This primitive requests the PHY to start the PHY PDU reception and carries all the control information needed for the corresponding PHY, such as start/end time of reception and PHY parameters for multiple bursts, types of preambles, etc.

8.1.3.23.2 Semantics of the service primitive

```
PHY_RXSTART.request
          (
          RXVECTOR
          )
```

8.1.3.23.3 When generated

This primitive is generated by the MAC to initiate PHY PDU reception.

8.1.3.23.4 Effect of receipt

PHY sends a confirmation primitive and at the proper moment enters the acquisition mode searching for a transmission with the proper PHY burst parameters.

8.1.3.24 PHY_RXSTART.confirmation

8.1.3.24.1 Function

This primitive confirms reception of PHY_RXSTART.request primitive.

8.1.3.24.2 Semantics of the service primitive

```
PHY_RXSTART.confirmation
          (
          )
```

8.1.3.24.3 When generated

This primitive is generated after reception of PHY_RXSTART.request primitive.

8.1.3.24.4 Effect of receipt

None.

8.1.3.25 PHY_RXSTART.indication

8.1.3.25.1 Function

This primitive indicates to the MAC the actual start of PHY PDU reception.

8.1.3.25.2 Semantics of the service primitive

```
PHY_RXSTART.indication
        (
        )
```

8.1.3.25.3 When generated

This primitive is generated after the actual start of PHY PDU reception.

8.1.3.25.4 Effect of receipt

None.

8.1.3.26 PHY_MACPDU.indication

8.1.3.26.1 Function

This primitive indicates an arrival of a MAC Message (DATA parameter).

8.1.3.26.2 Semantics of the service primitive

```
PHY_MACPDU.indication
        (
        DATA
        )
```

8.1.3.26.3 When generated

The primitive is generated after an arrival of a MAC Message from PHY.

8.1.3.26.4 Effect of receipt

None.

8.1.3.27 PHY_RXEND.indication

8.1.3.27.1 Function

This primitive indicates the moment of the PHY PDU end in the air.

8.1.3.27.2 Semantics of the service primitive

```
PHY_RXEND.indication
        (
        )
```

8.1.3.27.3 When generated

It is generated at the PHY PDU end in the air.

8.1.3.27.4 Effect of receipt

None.

8.1.3.28 Data transfer related primitives example

Figure 105 and Figure 106 illustrate possible scenarios for invocation of data transfer related primitives. Depending on the PHY interface with the MAC, MAC PDU transmission may occur after the accumulation of several MAC PDUs due to PHY specific buffering restrictions (e.g., FEC codeword length). In these cases, transmission events and reception events in Figure 105 should be interpreted as actual queuing events, while transmission and reception occurs according to transmission buffer and reception buffer status.

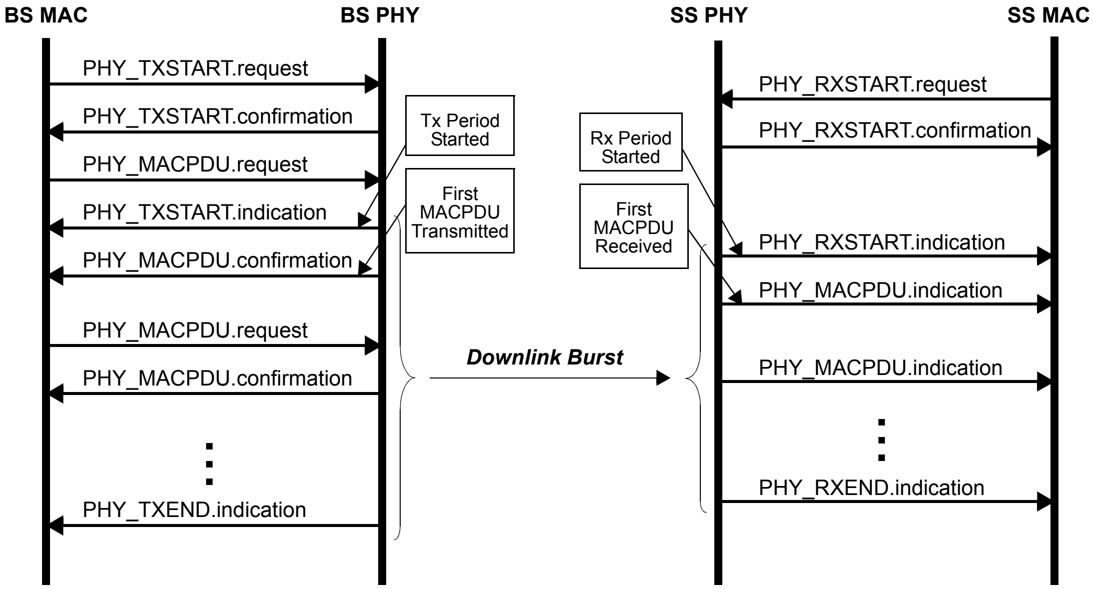


Figure 105—Example of usage of primitives during downlink PHY PDU transfer

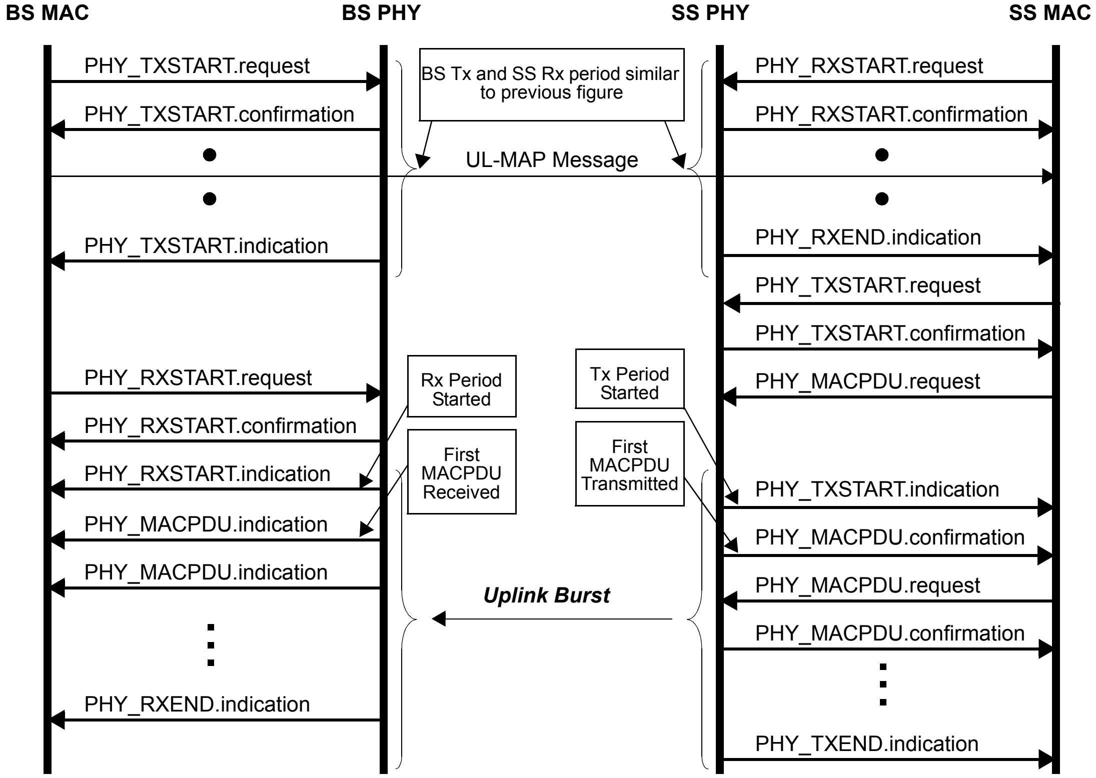


Figure 106—Example of usage of primitives during uplink PHY PDU transfer

8.2 PHY for 10–66 GHz

8.2.1 Overview

This PHY specification, targeted for operation in the 10–66 GHz frequency band, is designed with a high degree of flexibility in order to allow service providers the ability to optimize system deployments with respect to cell planning, cost, radio capabilities, services, and capacity.

In order to allow for flexible spectrum usage, both time division duplex (TDD) and frequency division duplex (FDD) configurations (8.2.4) are supported. Both cases use a burst transmission format whose framing mechanism (8.2.5.1) supports adaptive burst profiling in which transmission parameters, including the modulation and coding schemes, may be adjusted individually to each subscriber station (SS) on a frame-by-frame basis. The FDD case supports full-duplex SSs as well as half-duplex SSs, which do not transmit and receive simultaneously.

The uplink PHY is based on a combination of time division multiple access (TDMA) and demand assigned multiple access (DAMA). In particular, the uplink channel is divided into a number of time slots. The number of slots assigned for various uses (registration, contention, guard, or user traffic) is controlled by the MAC layer in the BS and may vary over time for optimal performance. The downlink channel is time division multiplexed (TDM), with the information for each subscriber station multiplexed onto a single stream of data and received by all subscriber stations within the same sector. To support half-duplex FDD subscriber stations, provision is also made for a TDMA portion of the downlink.

The downlink PHY includes a transmission CS that inserts a pointer byte at the beginning of the payload to help the receiver identify the beginning of a MAC PDU. Data bits coming from the transmission CS are randomized, FEC encoded, and mapped to a QPSK, 16-QAM, or 64-QAM (optional) signal constellation.

The uplink PHY is based upon TDMA burst transmission. Each burst is designed to carry variable-length MAC PDUs. The transmitter randomizes the incoming data, FEC encodes it, and maps the coded bits to a QPSK, 16-QAM (optional), or 64-QAM (optional) constellation.

8.2.2 PHY SAP parameter definitions

This subclause defines the PHY SAP parameters (8.1.3) used in this physical layer specification.

8.2.2.1 SCHED_PARAM_VECTOR

The SCHED_PARAM_VECTOR parameters and values are shown in Table 75.

Table 75—SCHED_PARAM_VECTOR for 10–66 GHz PHY

Parameter	Value
Symbol Rate	16 to 40 (in Mbaud)
Modulation density	2, 4, or 6
FEC block size	0 to 511 bytes
FEC payload	0 to 255 bytes
Uplink Preamble length	16 or 32 Symbols
PHY Overhead	0 to 256 Symbols

8.2.2.2 DCD_PARAM_VECTOR

The DCD_PARAM_VECTOR parameters and values are shown in Table 76.

Table 76—DCD_PARAM_VECTOR for 10–66 GHz PHY

Parameter	Value
RF Channel number	0 to maximum number of channels allowed in the system
Symbol Rate	16 to 40 (in Mbaud)
Number of active PHY burst profiles	1-13
Start active region in frame	0-65535 (in symbols)
End active region in frame	0-65535 (in symbols)

8.2.2.3 UCD_PARAM_VEC

The UCD_PARAM_VEC parameters and values are shown in Table 77.

Table 77—UCD_PARAM_VEC for 10–66 GHz PHY

Parameter	Value
RF Channel number	0 to maximum number of channels allowed in the system
Symbol Rate	16 to 40 (in Mbaud)
Number of active PHY burst profiles	3-12
Start active region in frame	0-65535 (in symbols)
End active region in frame	0-65535 (in symbols)

8.2.2.4 RNG_REQ_VECTOR

The RNG_REQ_VECTOR parameters and values are shown in Table 78.

Table 78—RNG_REQ_VECTOR for 10–66 GHz PHY

Parameter	Value
Frequency change adjustment	(-1000…+1000) (in KHz)
Time alignment	(-32768…+32767) in quarter symbols
Power adjust	(-128…+127) number of 0.5 dB

8.2.2.5 RNG_IND_VECTOR

The RNG_IND_VECTOR parameters and values are shown in Table 79.

Table 79—RNG_IND_VECTOR for 10–66 GHz PHY

Parameter	Value
Frequency deviation	(-1000…+1000) In KHz
RSSI	(0 to RSSI_MAX) (in dB)
Relative symbol time deviation	-16…15 (in quarter symbols)
Receiver failure	(0-OK, 1-failure)

8.2.2.6 TXVECTOR

The TXVECTOR parameters and values are shown in Table 80.

Table 80—TXVECTOR for 10–66 GHz PHY

Parameter	Value
Symbol Rate	16 to 40 (in Mbaud)
Burst profile used	0-15
Start transmit in frame	0-65535 (in symbols)
End transmit in frame	0-65535 (in symbols)
Actual number of bytes transmitted	0-65535 (in bytes)

8.2.2.7 TXSTATUS

The TXSTATUS parameters and values are shown in Table 81.

Table 81—TXSTATUS for 10–66 GHz PHY

Parameter	Value
Overrun	(0-65535) in symbols, 0 indicates no-overrun
Underrun	(0-65535) in symbols, 0 indicates no-underrun

8.2.2.8 RXVECTOR

The RXVECTOR parameters and values are shown in Table 82.

Table 82—RXVECTOR for 10–66 GHz PHY

Parameter	Value
Symbol Rate	16 to 40 (in Mbaud)
Burst profile used	0-15
Start Receive in frame	0-65535 (in symbols)
End Receive in frame	0-65535 (in symbols)
Actual number of bytes expected	0-65535 (in bytes)

8.2.2.9 RXSTATUS

The RXSTATUS parameters and values are shown in Table 83.

Table 83—RXSTATUS for 10–66 GHz PHY

Parameter	Value
RSSI level	(0 to RSSI_MAX) in dB
Number of bytes received	(0-65535) in bytes
Relative symbol time deviation	-16…15 (in quarter symbols)
Estimated number of byte errors	(0-65535) in bytes
Overrun	(0-65535) in symbols, 0 indicates no-overrun
Underrun	(0-65535) in symbols, 0 indicates no-underrun
Integrity	0 (valid data), 1 (invalid data)

8.2.3 Framing

This PHY specification operates in a framed format (6.2.7). Within each frame are a downlink subframe and an uplink subframe. The downlink subframe begins with information necessary for frame synchronization and control. In the TDD case, the downlink subframe comes first, followed by the uplink subframe. In the FDD case, uplink transmissions occur concurrently with the downlink frame.

Each SS shall attempt to receive all portions of the downlink except for those bursts whose burst profile is either not implemented by the SS or is less robust than the SS's current operational downlink burst profile. Half-duplex SSs shall not attempt to listen to portions of the downlink coincident with their allocated uplink transmission, if any, adjusted by their Tx time advance.

8.2.3.1 Supported frame durations

Table 84 indicates the supported frame durations.

Table 84—Frame durations and frame duration codes

Frame duration code	Frame duration (T_F)	Units
0x01	0.5	ms
0x02	1	ms
0x03	2	ms

8.2.4 Duplexing techniques and PHY Type parameter encodings

Both frequency division and time division duplexing are supported. The duplexing method shall be reflected in the PHY Type parameter (11.1.2.1) as shown in Table 85.

Table 85—PHY Type parameter encoding

PHY Type	Value
TDD	0
FDD	1

8.2.4.1 FDD operation

In FDD operation, the uplink and downlink channels are on separate frequencies. The capability of the downlink to be transmitted in bursts facilitates the use of different modulation types and allows the system to simultaneously support full-duplex subscriber stations (which can transmit and receive simultaneously) and half-duplex subscriber stations (which do not). Note that the downlink carrier may be continuous, as demonstrated in Figure 107 (third frame). Figure 107 describes the basics of the FDD operation.

In the case of a half-duplex SS, transition gaps, as described in 8.2.4.2.1 and 8.2.4.2.2, apply.

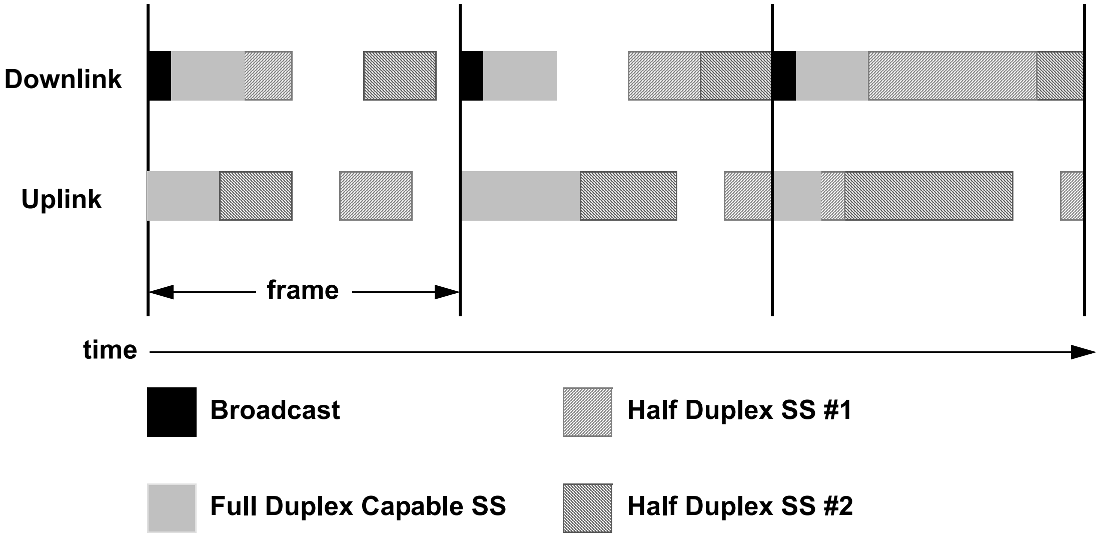


Figure 107—Example of FDD bandwidth allocation

8.2.4.2 TDD operation

In the case of TDD, the uplink and downlink transmissions share the same frequency but are separated in time, as shown in Figure 108. A TDD frame also has a fixed duration and contains one downlink and one uplink subframe. The TDD framing is adaptive in that the link capacity allocated to the downlink versus the uplink may vary.

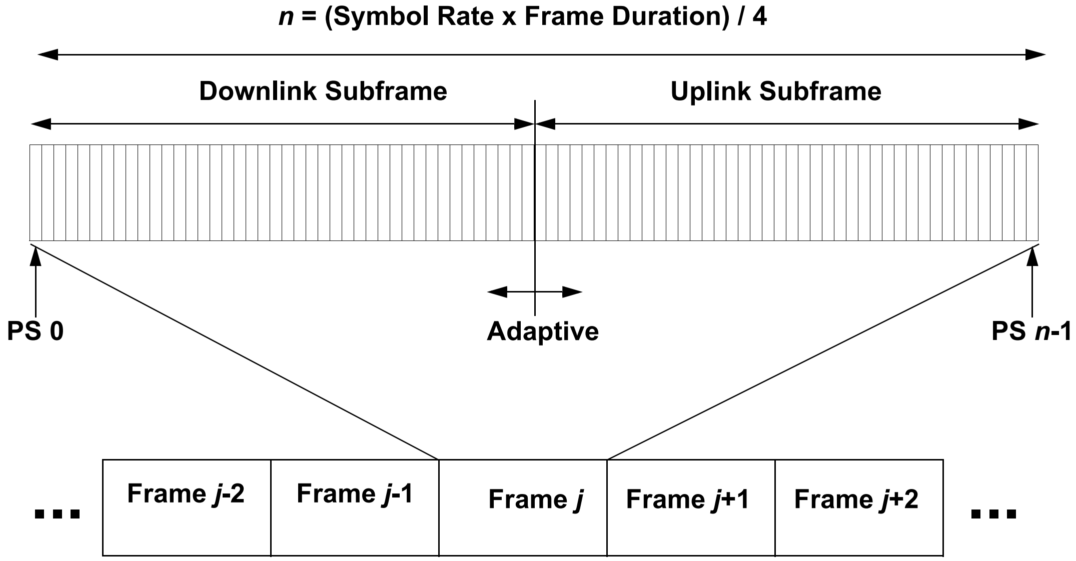


Figure 108—TDD frame structure

8.2.4.2.1 Tx/Rx Transition Gap

The Tx/Rx Transition Gap (TTG) is a gap between the downlink burst and the subsequent uplink burst. This gap allows time for the BS to switch from transmit to receive mode and SSs to switch from receive to transmit mode. During this gap, the BS and SS are not transmitting modulated data but simply allowing the BS transmitter carrier to ramp down, the Tx/Rx antenna switch to actuate, and the BS receiver section to activate. After the gap, the BS receiver shall look for the first symbols of uplink burst. This gap is an integer number of physical slots (PSs) durations and starts on a PS boundary.

8.2.4.2.2 Rx/Tx Transition Gap

The Rx/Tx Transition Gap is a gap between the uplink burst and the subsequent downlink burst. This gap allows time for the BS to switch from receive to transmit mode and SSs to switch from transmit to receive mode. During this gap, the BS and SS are not transmitting modulated data but simply allowing the BS transmitter carrier to ramp up, the Tx/Rx antenna switch to actuate, and the SS receiver sections to activate. After the gap, the SS receivers shall look for the first symbols of QPSK modulated data in the downlink burst. This gap is an integer number of PSs durations and starts on a PS boundary.

8.2.5 Downlink PHY

The available bandwidth in the downlink direction is defined with a granularity of one PS. The available bandwidth in the uplink direction is defined with a granularity of one minislot, where the minislot length is 2^m PSs (m ranges from 0 through 7). The number of PSs with each frame is a function of the symbol rate. The symbol rate is selected in order to obtain an integral number of PSs within each frame. For example, with a 20 Mbaud symbol rate, there are 5000 PSs within a 1 ms frame.

8.2.5.1 Downlink subframe

The structure of the downlink subframe using TDD is illustrated in Figure 109. The downlink subframe begins with a Frame Start Preamble used by the PHY for synchronization and equalization. This is followed by the frame control section, containing downlink and uplink maps stating the PSs at which bursts begin. The following TDM portion carries the data, organized into bursts with different burst profiles and therefore different level of transmission robustness. The bursts are transmitted in order of decreasing robustness. For example, with the use of a single FEC type with fixed parameters, data begins with QPSK modulation, followed by 16-QAM, followed by 64-QAM. In the case of TDD, a TTG separates the downlink subframe from the uplink subframe.

Each SS receives and decodes the control information of the downlink and looks for MAC headers indicating data for that SS in the remainder of the downlink subframe.

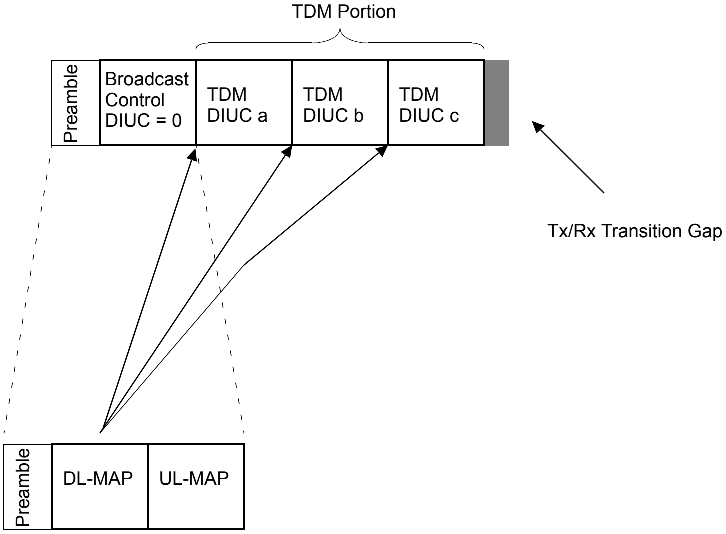


Figure 109—TDD downlink subframe structure

In the FDD case, the structure of the downlink subframe is illustrated in Figure 110. Like the TDD case, the downlink subframe begins with a Frame Start Preamble followed by a frame control section and a TDM portion organized into bursts transmitted in decreasing order of burst profile robustness. This TDM portion of the downlink subframe contains data transmitted to one or more of the following:

— full-duplex SSs

— half-duplex SSs scheduled to transmit later in the frame than they receive

— half-duplex SSs not scheduled to transmit in this frame.

The FDD downlink subframe continues with a TDMA portion used to transmit data to any half-duplex SSs scheduled to transmit earlier in the frame than they receive. This allows an individual SS to decode a specific portion of the downlink without the need to decode the entire downlink subframe. In the TDMA portion, each burst begins with the Downlink TDMA Burst Preamble for phase resynchronization. Bursts in the TDMA portion need not be ordered by burst profile robustness. The FDD frame control section includes a map of both the TDM and TDMA bursts.

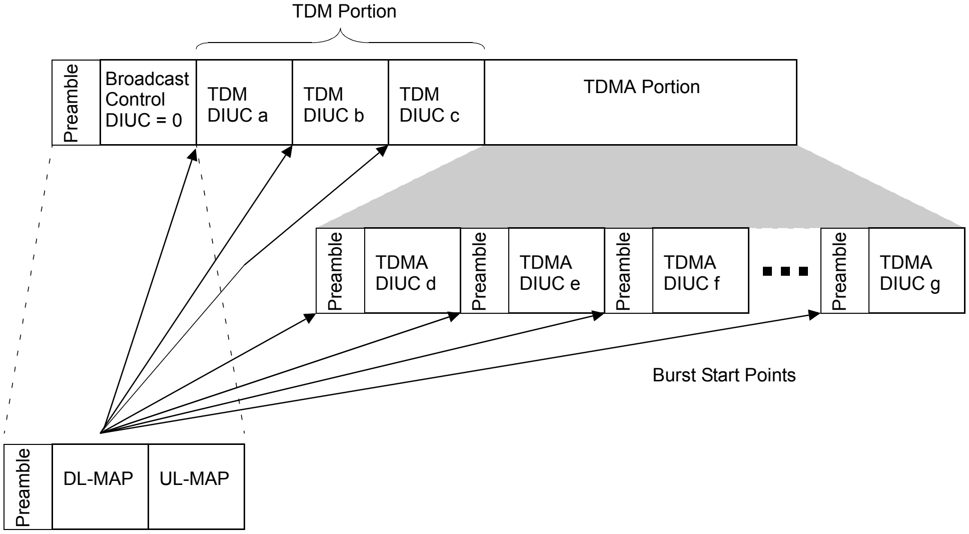


Figure 110—FDD downlink subframe structure

The TDD downlink subframe, which inherently contains data transmitted to SSs that transmit later in the frame than they receive, is identical in structure to the FDD downlink subframe for a frame in which no half-duplex SSs are scheduled to transmit before they receive.

8.2.5.1.1 Downlink burst preambles

As shown in Table 86, two downlink burst preambles are used. The Frame Start Preamble shall begin each downlink frame. The Downlink TDMA Burst Preamble shall begin each TDMA burst in the TDMA portion of the downlink subframe.

Table 86—Downlink burst preambles

Preamble name	Burst profile	Preamble Type	Modulation Type
Frame Start Preamble	TDM Burst	1	QPSK
Downlink TDMA Burst Preamble	TDMA Burst	2	QPSK

Both preambles use QPSK modulation and are based upon +45 degrees rotated constant amplitude zero auto-correlation (CAZAC) sequences (Milewski [B15]). The amplitude of the preamble shall depend on the downlink power adjustment rule (8.2.5.4.7). In the case of the constant peak power scheme (power adjustment rule=0), the preamble shall be transmitted such that its constellation points coincide with the outermost constellation points of the modulation scheme in use. In the case of the constant mean power scheme (power adjustment rule=1), it shall be transmitted with the mean power of the constellation points of the modulation scheme in use.

The Frame Start Preamble (Table 87) consists of a 32-symbol sequence generated by repeating a 16 symbol CAZAC sequence. The Downlink TDMA Burst Preamble (Table 88) consists of a 16 symbol sequence generated by repeating an 8-symbol CAZAC sequence.

Table 87—Frame start preamble

Symbol	I	Q	B(1)	B(2)
1 and 17	1	1	0	0
2 and 18	–1	1	1	0
3 and 19	–1	–1	1	1
4 and 20	1	–1	0	1
5 and 21	1	1	0	0
6 and 22	–1	–1	1	1
7 and 23	1	1	0	0
8 and 24	–1	–1	1	1
9 and 25	1	1	0	0
10 and 26	1	–1	0	1
11 and 27	–1	–1	1	1
12 and 28	–1	1	1	0
13 and 29	1	1	0	0
14 and 30	1	1	0	0
15 and 31	1	1	0	0
16 and 32	1	1	0	0

Table 88—Downlink TDMA burst preamble

Symbol	I	Q	B(1)	B(2)
1 and 9	1	1	0	0
2 and 10	1	1	0	0
3 and 11	1	1	0	0
4 and 12	–1	1	1	0
5 and 13	–1	–1	1	1
6 and 14	1	1	0	0
7 and 15	–1	–1	1	1
8 and 16	–1	1	1	0

8.2.5.1.2 Frame control section

The frame control section is the first portion of the downlink frame following the preamble. It is used for control information destined for all SSs. This control information shall not be encrypted. The information transmitted in this section always uses the well-known downlink burst profile with DIUC=0.

The frame control section shall contain a DL-MAP message (6.2.2.3.2) for the channel followed by one UL-MAP message (6.2.2.3.4) for each associated uplink channel. In addition, it may contain DCD and UCD messages (6.2.2.3.1 and 6.2.2.3.3) following the last UL-MAP message. No other messages shall be sent in the frame control section.

8.2.5.1.2.1 DL-MAP elements

The information elements as defined in Table 89 follow the Number of DL-MAP Elements field of the DL-MAP message, as described in 6.2.2.3.2. The map information elements shall be in chronological order. Note that this is not necessarily DIUC order (as DIUC numbering does not necessarily reflect robustness of the burst profile) or CID order.

Table 89—DL_MAP_Information_Element

Syntax	Size	Notes
DL_MAP_Information_Element() {		
DIUC	4	
StartPS	16	the starting point of the burst, in units of PS where the first PS in a given frame has StartPS=0
}		

8.2.5.1.2.2 DL-MAP PHY synchronization field definition

The format of the PHY Synchronization Field of the DL-MAP message, as described in 6.2.2.3.2, is given in Table 90. The Frame Duration Codes are given in Table 84. The Frame Number is incremented by 1 each frame and eventually wraps around to zero.

Table 90—PHY synchronization field

Syntax	Size	Notes
PHY Synchronization Field() {		
Frame Duration Code()	8 bits	
Frame Number	24 bits	
}		

8.2.5.1.2.3 UL-MAP allocation start time definition

The allocation start time (Alloc Start Time) is the effective start time of the uplink allocation defined by the UL-MAP in units of mini-slots. The start time is relative to the start of the frame in which the UL-MAP message is transmitted.

8.2.5.1.2.4 Required DCD parameters

The following parameters shall be included in the DCD message:

- **BS Transmit Power** [Note: to be used by SSs to validate radio link conditions]
- **PHY type**
- **FDD/TDD frame duration**

8.2.5.1.2.5 Downlink_Burst_Profile

Each Downlink_Burst_Profile in the DCD message (6.2.2.3.1) shall include the following parameters:

- Modulation type
- FEC Code Type
- Last codeword length
- DIUC mandatory exit threshold
- DIUC minimum entry threshold
- Preamble Presence

If the FEC Code Type is 1, 2, or 3 (RS codes), the Downlink_Burst_Profile shall also include

- RS information bytes (K)
- RS parity bytes (R)

If the FEC Code Type is 2, the Downlink_Burst_Profile shall also include

- BCC code type

If the FEC Code Type is 4, the Downlink_Burst_Profile shall also include

- BTC row code type
- BTC column code type
- BTC interleaving type

The mapping between Burst Profile and DIUC is given in Table 91.

The Downlink Burst Profile 1 (DIUC=0) parameters defined in 8.2.5.4.5 shall be stored in the SS and shall not be included in the DCD message.

The Gap Downlink Burst Profile (DIUC=14) indicates a silent interval in downlink transmission. It is well-known and shall not be defined in the DCD message.

Table 91—Mapping of burst profile to DIUC

Burst profile	DIUC
Downlink Burst Profile 1	0
Downlink Burst Profile 2	1
Downlink Burst Profile 3	2
Downlink Burst Profile 4	3
Downlink Burst Profile 5	4
Downlink Burst Profile 6	5
Downlink Burst Profile 7	6
Downlink Burst Profile 8	7
Downlink Burst Profile 9	8
Downlink Burst Profile 10	9
Downlink Burst Profile 11	10
Downlink Burst Profile 12	11
Downlink Burst Profile 13	12
reserved	13
Gap	14
End of DL-MAP	15

The End of DL-MAP Burst Profile (DIUC=15) indicates the first PS after the end of the DL subframe. It is well known and shall not be included in the DCD message.

Table 92 defines the format of the Downlink_Burst_Profile, which is used in the DCD message (6.2.2.3.1). The Downlink_Burst_Profile is encoded with a Type of 1, an 8-bit length, and a 4-bit DIUC. The DIUC field is associated with the Downlink Burst Profile and Thresholds. The DIUC value is used in the DL-MAP message to specify the Burst Profile to be used for a specific downlink burst.

Table 92—Downlink_Burst_Profile format

Syntax	Size	Notes
Type=1	8 bits	
Length	Variable	
reserved	4 bits	shall be set to zero
DIUC	4 bits	
TLV encoded information	Variable	TLV Specific

8.2.5.2 Downlink burst allocation

The downlink data sections are used for transmitting data and control messages to the specific SSs. The data are always FEC coded and are transmitted at the current operating modulation of the individual SS. In the TDM portion, data shall be transmitted in order of decreasing burst profile robustness. In the case of a TDMA portion, the data are grouped into separately delineated bursts that need not be in robustness order (see 8.2.5.1). The DL-MAP message contains a map stating at which PS the burst profile changes occur. If the downlink data does not fill the entire downlink subframe, the transmitter is shut down. FEC codewords within a burst are arranged in a compact form aligned to bit-level boundaries. This implies that, while the first FEC codeword shall start on the first PS boundary, succeeding FEC codewords may start even within a modulation symbol or within a PS if the succeeding FEC codeword ended within a modulation symbol or within a PS. The exact alignment conditions depend on the burst profile parameters.

In the case of shortening the last FEC block within a burst (optional, see 11.1.2.2), the downlink map provides an implicit indication.

In general, the number of PSs *i* (which shall be an integer) allocated to a particular burst can be calculated from the downlink MAP, which indicates the starting position of each burst as well as the burst profiles. Let *n* denote the minimum number of PSs required for one FEC codeword of the given burst profile (note that *n* is not necessarily an integer). Then $i=kn+j+q$, where *k* is the number of whole FEC codewords that fit in the burst, *j* (not necessarily an integer) is the number of PSs occupied by the largest possible shortened codeword, and *q* ($0 \leq q < 1$) is the number of PSs occupied by pad bits inserted at the end of the burst to guarantee that *i* is an integer. In Fixed Codeword Operation (8.2.5.4.4.1), *j* is always 0. Recall that a codeword can end partway through a modulation symbol as well as partway through a PS. When this occurs, the next codeword shall start immediately, with no pad bits inserted. At the end of the burst (i.e., when there is no next codeword), then 4*q* symbols are added as padding (if required) to complete the PS allocated in the downlink map. The number of padding bits in these padding symbols is 4*q* times the modulation density, where the modulation density is 2 for QPSK, 4 for 16-QAM, and 6 for 64-QAM. Note that padding bits may be required with or without shortening. Either *k* or *j*, but not both, may be zero. The number *j* implies some number of bits *b*. Assuming *j* is nonzero, it shall be large enough such that *b* is larger than the number of FEC bits, *r*, added by the FEC scheme for the burst. The number of bits (preferably an integral number of bytes) available for user data in the shortened FEC codeword is *b–r*. Any bits that may be left over from a fractional byte are encoded as binary 1 to ensure compatibility with the choice of 0xFF for pad. A codeword cannot have less than 6 information bytes. This is illustrated in Figure 111.

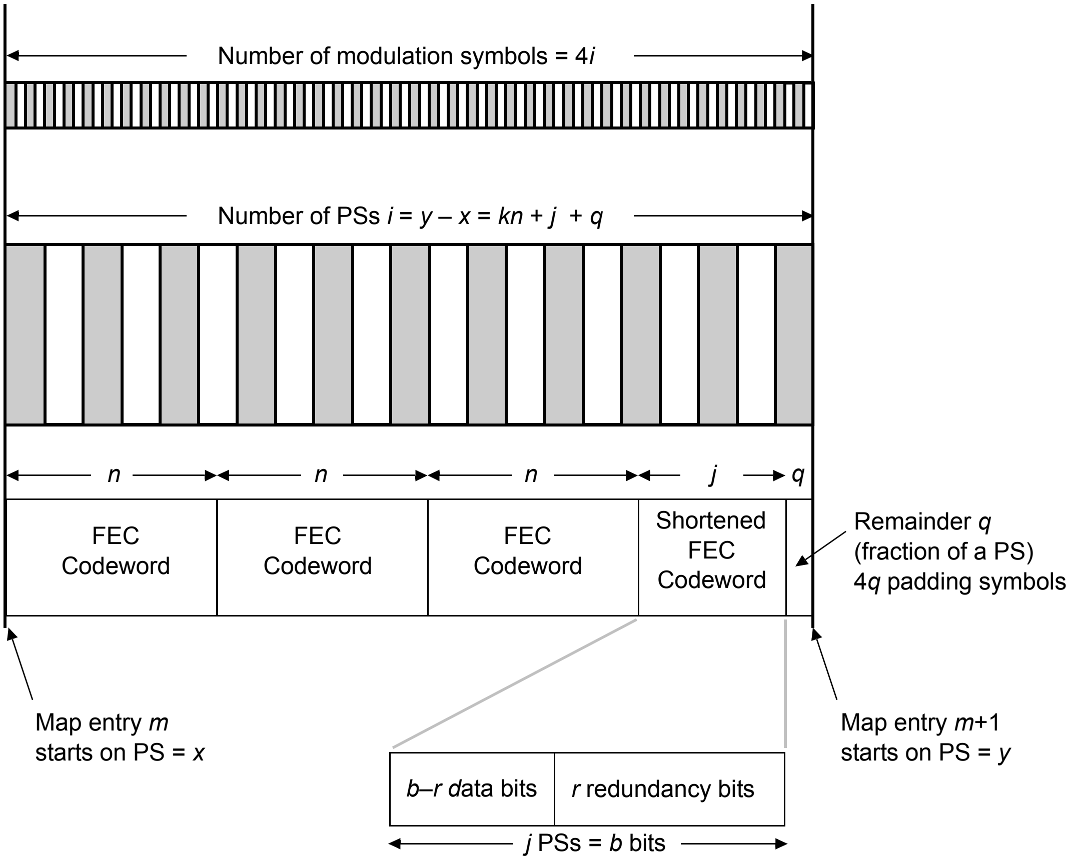

Number of modulation symbols = 4*i*
Number of PSs *i* = *y* – *x* = *kn* + *j* + *q*
n
n
n
j
q
FEC Codeword
FEC Codeword
FEC Codeword
Shortened FEC Codeword
Remainder *q* (fraction of a PS) 4*q* padding symbols
Map entry *m* starts on PS = *x*
Map entry *m*+1 starts on PS = *y*
b–*r* data bits
r redundancy bits
j PSs = *b* bits

Figure 111—Downlink map usage with shortened FEC blocks—TDM case

In the case of TDMA downlink, a burst includes the Downlink TDMA Burst Preamble of length *p* PSs, and the downlink map entry points to its beginning (Figure 112).

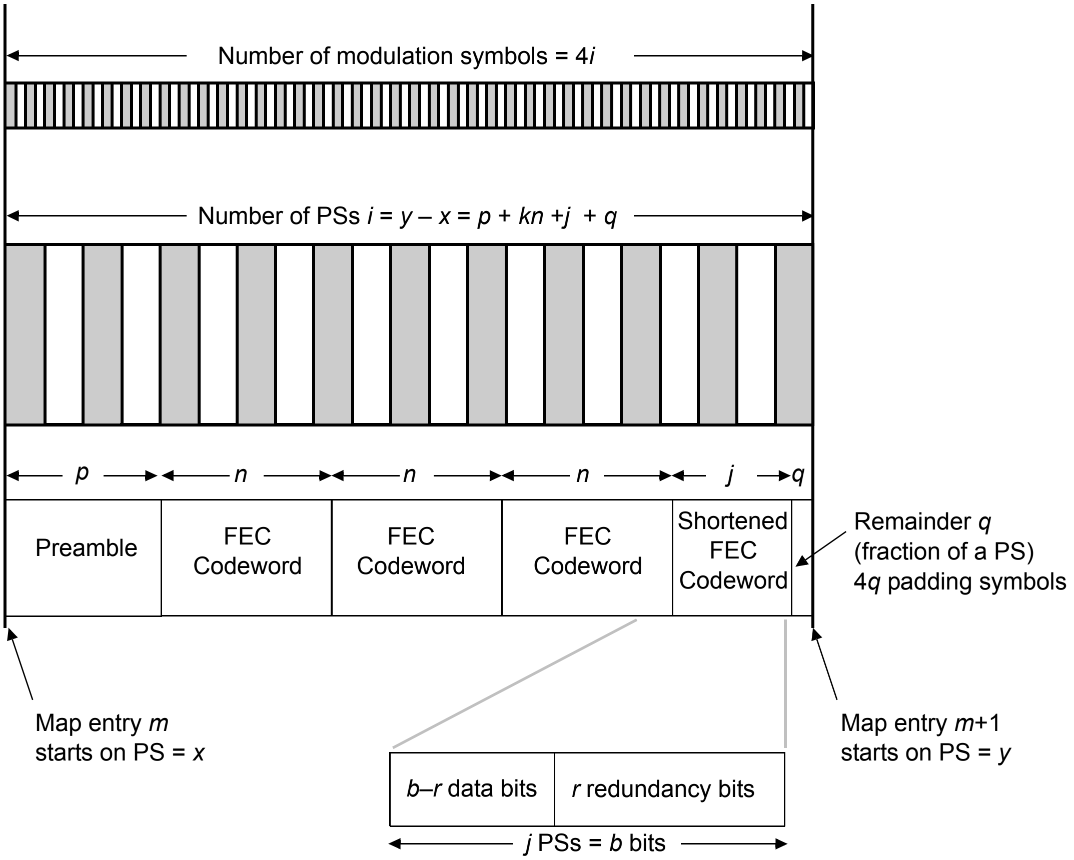

Number of modulation symbols = 4*i*
Number of PSs *i* = *y* – *x* = *p* + *kn* +*j* + *q*
p
n
n
n
j
q
Preamble
FEC Codeword
FEC Codeword
FEC Codeword
Shortened FEC Codeword
Remainder *q* (fraction of a PS) 4*q* padding symbols
Map entry *m* starts on PS = *x*
Map entry *m*+1 starts on PS = *y*
b–*r* data bits
r redundancy bits
j PSs = *b* bits

Figure 112—Downlink map usage with shortened FEC blocks—TDMA case

8.2.5.3 Downlink transmission CS

The downlink payload shall be segmented into blocks of data designed to fit into the proper codeword size after the CS pointer byte is added. Note that the payload length may vary, depending on whether shortening of codewords is allowed or not for this burst profile. A pointer byte shall be added to each payload segment, as illustrated in Figure 113.

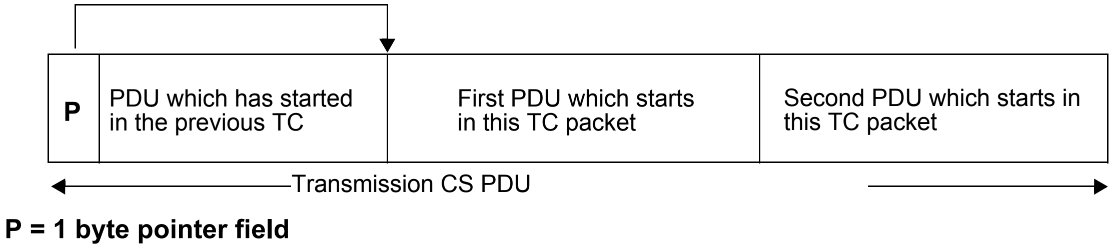

P
PDU which has started in the previous TC
First PDU which starts in this TC packet
Second PDU which starts in this TC packet
Transmission CS PDU
P = 1 byte pointer field

Figure 113—Format of the downlink transmission CS PDU

The pointer field identifies the byte number in the packet which indicates either the beginning of the first MAC PDU to start in the packet or the beginning of any stuff bytes that precede the next MAC PDU. For reference, the first byte in the packet is referred to as byte number 1. If no MAC PDU or stuff bytes begin in the CS packet, then the pointer byte is set to 0. When no data is available to transmit, a stuff_byte pattern having a value (0xFF) shall be used within the payload to fill any gaps between the IEEE Std 802.16-2001 MAC PDUs. This value is chosen as an unused value for the first byte of the IEEE Std 802.16-2001 MAC PDU, which is designed to never have this value.

8.2.5.4 Downlink physical medium dependent (PMD) sublayer

The downlink physical layer coding and modulation for this mode is summarized in the block diagram in Figure 114.

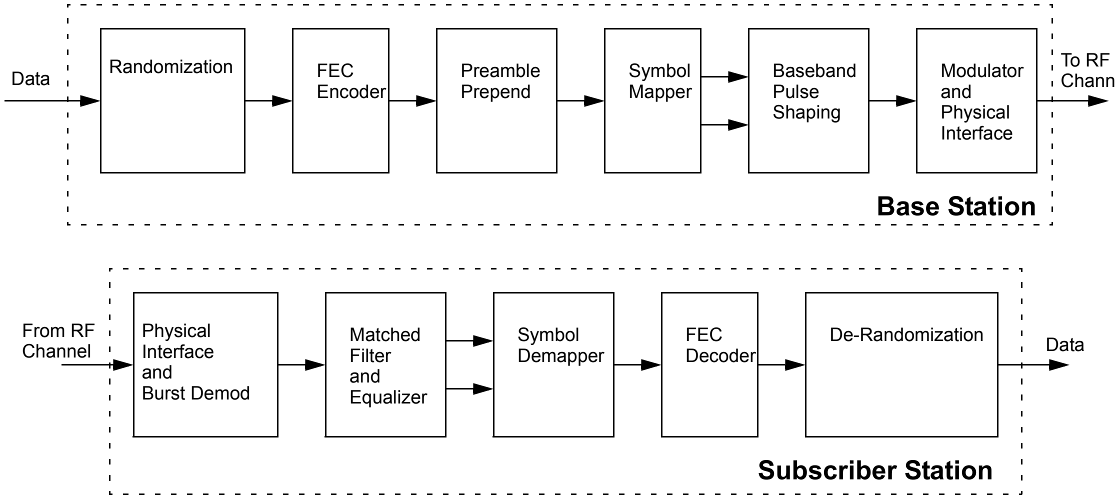


Figure 114—Conceptual block diagram of the downlink PMD sublayer

8.2.5.4.1 Burst profile definitions

The downlink channel supports adaptive burst profiling on the user data portion of the frame. Up to twelve burst profiles can be defined. The parameters of each are communicated to the SSs via MAC messages during the frame control section of the downlink frame (see 8.2.5.1). The downlink channel and burst profiles are communicated to the SSs via the MAC messages described in 6.2.2.3.1.

The use of DIUCs shall be constrained as shown in Table 93.

Table 93—DIUC allocation

DIUC	Usage
0	frame control (well known, not in DCD message)
1-6	TDM Burst Profiles (no preamble)
7-12	TDMA Burst Profiles (preamble prefixed)
13	*reserved*
14	Gap (well known, not in DCD message)
15	End of Map

8.2.5.4.2 Downlink PHY SS capability set parameters

Since there are optional modulation and FEC schemes that can be implemented at the SS, a method for identifying the capability to the BS is required (i.e., including the highest order modulation supported, the optional FEC coding schemes supported, and the minimum shortened last codeword length supported). This information shall be communicated to the BS during the subscriber registration period.

8.2.5.4.3 Randomization

Randomization shall be employed to minimize the possibility of transmission of an unmodulated carrier and to ensure adequate numbers of bit transitions to support clock recovery. The stream of downlink packets shall be randomized by modulo-2 addition of the data with the output of the pseudorandom binary sequence generator, as illustrated in Figure 115. The generator polynomial for the pseudorandom binary sequence shall be $c(x) = x^{15} + x^{14} + 1$.

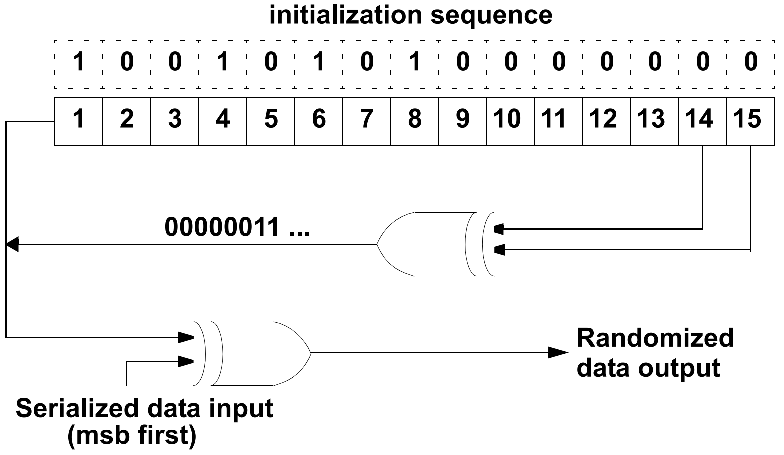


Figure 115—Randomizer logic diagram

At the beginning of each burst, the pseudorandom binary sequence register is cleared and the seed value of 100101010000000 is loaded. A burst corresponds to either a TDM burst beginning with the Frame Start Preamble or a TDMA burst beginning with a Downlink TDMA Burst Preamble (8.2.5.1.1). The preambles are not randomized. The seed value shall be used to calculate the randomization bit, which is combined in an XOR with the first bit of data of each burst. The randomizer sequence is applied only to information bits.

8.2.5.4.4 Downlink forward error correction

The forward error correction (FEC) schemes are selectable from the types in Table 94.

Table 94—FEC Code Types

Code Type	Outer Code	Inner Code
1	Reed-Solomon over GF(256)	None
2	Reed-Solomon over GF(256)	(24,16) Block convolutional code
3 (Optional)	Reed-Solomon over GF(256)	(9,8) Parity check code
4 (Optional)	Block Turbo Code	—

Implementation and use of Code Types 3 and 4 is optional. Code Types 1 and 2 shall be implemented by all BSs and SSs. Code Type 2 shall not be used except in the case of QPSK modulation. In the case of QPSK, any of the four Code Types may be used, with one exception: Code Type 2 shall always be used for the control channel (DIUC=0).

Following is a summary of the four Code Types:

a) ***Code Type 1:*** *Reed-Solomon only:* This case is useful either for a large data block or when high coding rate is required. The protection could vary between t=0 to t=16.

b) ***Code Type 2:*** *Reed-Solomon + Block convolutional code (soft decodable):* This case is useful for low to moderate coding rates providing good carrier-to-noise ratio (C/N) enhancements. The coding rate of the inner block convolutional code (BCC) is 2/3. Note: The number of information bytes shall be even in this case.

c) ***Code Type 3:*** *Reed-Solomon + Parity check:* This optional code is useful for moderate to high coding rates with small to medium size blocks (i.e., K = 16, 53 or 128). The code itself is a simple bit wise parity check operating on byte (8-bit) level. The parity code can be used for error correction, preferably employing a soft decoder.

d) ***Code Type 4:*** *Block Turbo Code:* This optional code is used to significantly lower the required C/I level needed for reliable communication, and can be used to either extend the range of a base station or increase the code rate for greater throughput.

8.2.5.4.4.1 Outer code for Code Types 1-3, downlink

The outer block code for Code Types 1-3 shall be a shortened, systematic Reed-Solomon code generated from GF(256) with information block length K variable from 6-255 bytes and error correction capability T able to correct from 0 to 16 byte errors. The specified code generator polynomials are given by:

Code Generator Polynomial: $g(\mathrm{x}) = (x+\mu^0)(x+\mu^1)(x+\mu^2) \ldots (x+\mu^{2T-1})$, where $\mu = 02_{\mathrm{hex}}$

Field Generator Polynomial: $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

The specified code has a block length of 255 bytes and shall be configured as an RS(255,255-*R*) code with information bytes preceded by (255-*N*) zero symbols, where *N* is the codeword length and *R* the number of redundancy bytes (*R* = 2**T* ranges from 0 to 32, inclusive).

The value of *K* and *T* are specified for each burst profile by the MAC. Both Fixed Codeword Operation and Shortened Last Codeword Operation, as defined below, are allowed.

When using Code Type 2, the number of information bytes *K* shall always be an even number so that the total codeword size (*K*+*R*) is also an even number. This is due to the fact that the BCC code requires a pair of bytes on which to operate.

a) Fixed Codeword Operation

In Fixed Codeword Operation, the number of information bytes *K* is the same in each Reed-Solomon codeword. If the MAC messages in a burst require fewer bytes than are carried by an integral number of codewords, stuff bytes (FF_{hex}) shall be added between MAC messages or after the last MAC message so that the total message length is an integral multiple of *K* bytes.

The SS determines the number of codewords in its downlink burst from the Downlink Map message, which defines the beginning point of each burst, and hence the length. The BS determines the number of codewords in the downlink as it scheduled this transmission event and is aware about its length. Using the burst length, both the SS and the BS calculate the number of full-length RS codewords that can be carried by each burst.

The process used by the BS to encode each burst is described below:

When the number of randomized MAC message bytes (*M*) entering the FEC process is less than *K* bytes, Operation A shall be performed:

A1) Add (*K*-*M*) stuff bytes (FF_{hex}) to the *M* byte block as a suffix.
A2) RS encode the *K* bytes and append the *R* parity bytes.
A3) Serialize the bytes and transmit them to the inner coder or the modulator MSB first.

When the number of randomized MAC message bytes (*M*) entering the FEC process is greater than or equal to *K* bytes, Operation B shall be performed:

B1) RS encode the first *K* bytes and append the *R* parity bytes.
B2) Subtract *K* from *M* (Let *M*=*M*–*K*).
B3) If the new *M* is greater than or equal to *K*, then repeat with the next set of bytes (go to B1).
B4) If the new *M* is zero, then stop; otherwise go to step A1 above and process the *M*<*K* case.

b) Shortened Last Codeword Operation

In the Shortened Last Codeword Operation, the number of information bytes in the final Reed-Solomon block of each burst is reduced from the normal number *K*, while the number of parity bytes *R* remains the same. The BS tailors the number of information bytes in the last codeword in order to minimize the number of stuff bytes to add to the end of the MAC message. The length of the burst is then set to the minimum number of PSs required to transport all of the burst's bytes, which include preamble, information, and parity bytes. The BS implicitly communicates the number of bytes in the shortened last codeword to the SS via the Downlink Map message, which defines the starting PS of each burst. The SS uses the Downlink Map information to calculate the number of full-length RS codewords and the length of the shortened last codeword that can be carried within the specified burst size. The BS performs a similar calculation as the SS for its encoding purposes.

To allow the receiving hardware to decode the previous Reed-Solomon codeword, no Reed-Solomon codeword shall have less than 6 information bytes. The number of information bytes carried by the shortened last codeword shall be between 6 and *K* bytes, inclusive. If the number of information bytes needing to be sent by the BS is less than 6 bytes of data, stuff bytes (FF_{hex}) shall be appended to the end of the data to bring the total number of information bytes up to the minimum of 6.

When using Code Type 2, the number of information bytes in the shortened last codeword shall always be an even number so that the total codeword size is also an even number. If an odd number of information bytes needs to be sent, a stuff byte (FF_{hex}) shall be appended to the end of the message to obtain an even number of bytes.

The process used by the BS to encode each burst is described below:

First, the full-sized Reed-Solomon codewords that precede the burst's final codeword are encoded as in the Fixed Codeword Mode above. The number of bytes allocated for the shortened last codeword by the Uplink Map is *k′* bytes, which shall be between 6 and *K* bytes. The remaining *M* bytes of the message are then encoded into these *k′* bytes using the following procedure:

A1) Add (*K*-*k′*) zero bytes to the *M* byte block as a prefix.
A2) RS encode the *K* bytes and append the *R* parity bytes.
A3) Discard all of the (*K*-*k′*) zero RS symbols.
A4) Serialize the bytes and transmit them to the inner coder or the modulator msb first.
A5) Perform the inner coding operation (if applicable).

8.2.5.4.4.2 Inner code for Code Type 2, downlink

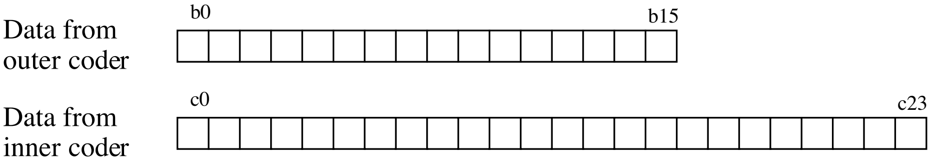
The inner code in Code Type 2 consists of short block codes derived from a 4-state, nonsystematic, punctured convolutional code (7,5). The trellis shall use the tail-biting method, where the last 2 bits of the message block are used to initialize the encoder memory, in order to avoid the overhead required for trellis termination. Thus, the encoder has the same initial and ending state for a message block.

For this concatenated coding scheme, the inner code message block is selected to be 16 bits. The puncturing pattern is described in Table 95 for the (24,16) case.

Table 95—Parameters of the inner codes for the BCC

Inner code rate	Puncture pattern G1 = 7, G2 = 5
2/3	11, 10

Figure 116 describes the exact encoding parity equations.



$c23=b15 \oplus b0 \oplus b1$

$c22=b15 \oplus b1$

$c21=b14 \oplus b15 \oplus b0$

$c20=b13 \oplus b14 \oplus b15$

$c19=b13 \oplus b15$

$c18=b12 \oplus b13 \oplus b14$

$c17=b11 \oplus b12 \oplus b13$

$c16=b11 \oplus b13$

$c15=b10 \oplus b11 \oplus b12$

$c14=b9 \oplus b10 \oplus b11$

$c13=b9 \oplus b11$

$c12=b8 \oplus b9 \oplus b10$

$c11=b7 \oplus b8 \oplus b9$

$c10=b7 \oplus b9$

$c9=b6 \oplus b7 \oplus b8$

$c8=b5 \oplus b6 \oplus b7$

$c7=b5 \oplus b7$

$c6=b4 \oplus b5 \oplus b6$

$c5=b3 \oplus b4 \oplus b5$

$c4=b3 \oplus b5$

$c3=b2 \oplus b3 \oplus b4$

$c2=b1 \oplus b2 \oplus b3$

$c1=b1 \oplus b3$

$c0=b0 \oplus b1 \oplus b2$

- 16 bits of data enter the inner BCC coder, b15 (msb) first.
- 24 bits of data exit the inner coder, c23 (msb) first.
- “$\oplus$” represents “XOR”

Figure 116—Inner code for Code Type 2 in the downlink

The number of information bytes shall be even since the BCC code operates on byte pairs.

8.2.5.4.4.3 Inner code for Code Type 3, downlink

For Code Type 3, a parity check bit is added to each Reed–Solomon (RS) symbol individually and inserted as the least significant bit (lsb) of the resulting 9-bit word. The parity is an XOR operation on all 8 bits within the symbol.

8.2.5.4.4.4 Code Type 4, downlink

Code Type 4, the Block Turbo Code (BTC), is a Turbo decoded Product Code (TPC). The idea of this coding scheme is to use extended Hamming block codes in a two-dimensional matrix. The two-dimensional code block is depicted in Figure 117. The k_x information bits in the rows are encoded into n_x bits, by using an extended Hamming binary block (n_x,k_x) code. Likewise, k_y information bits in the columns are encoded into n_y bits, by using the same or possibly different extended Hamming binary block (n_y,k_y) code. The resultant code block is comprised of multiple rows and columns of the constituent extended Hamming block codes.

For this standard, the rows shall be encoded first. After encoding the rows, the columns are encoded using another block code (n_y,k_y), where the check bits of the first code are also encoded. The overall block size of such a product code is $n = n_x$ x n_y, the total number of information bits k_x x k_y, the code rate is $R = R_x$ x R_y, where $R_i = k_i/n_i$ and i=x or y.

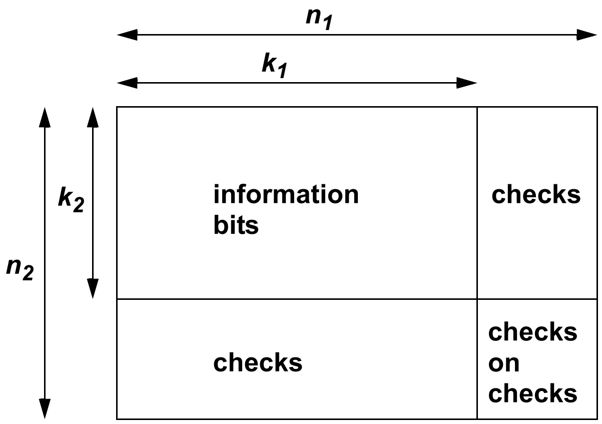


Figure 117—Two-dimensional product code matrix

Table 96 provides the generator polynomials of the constituent Hamming codes used in this specification.

Table 96—Hamming code generator polynomials

n	*k*	Generator polynomial
31	26	$x^5 + x^2 + 1$
63	57	$x^6 + x + 1$

The composite extended Hamming code specified requires addition of an overall even parity check bit at the end of each codeword.

The encoder for a BTC is composed of linear feedback shift registers (LFSRs), storage elements, and control logic. An example row (or column) encoder is shown here for clarification. The order of transmission is important so that the decoder may match for proper decoding. This specification mandates that the resultant code block be transmitted row by row, left to right, top to bottom, for the case when no interleaving is used (Interleaver Type 1 described below).

Figure 118 shows an example LFSR based on a $x^4 + x + 1$ Hamming code polynomial to encode a (15,11) Hamming code. Also shown is an even parity computation register that results in an extended Hamming code. Note that encoders for the required (64,57) and (32,26) codes follow the same design concept. This figure is shown for clarification of the BTC encoder design and does not depict an actual design implementation.

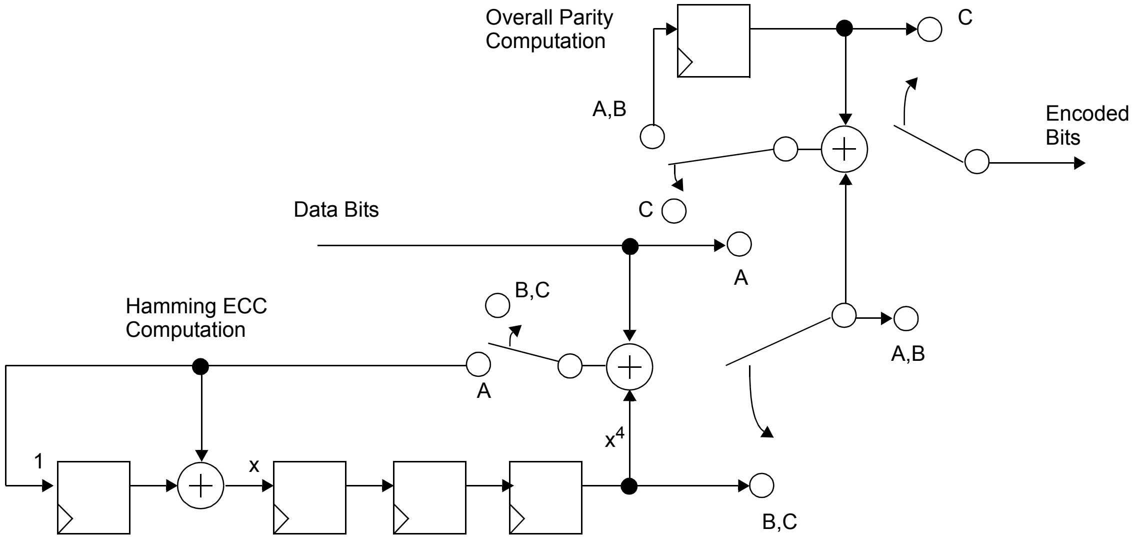


Figure 118—Example encoder for a (16,11) extended Hamming Code

The example circuit begins with all toggle switches in position A. Data to be encoded is fed as input one bit per clock (lsb first) to both the Hamming error correction code (ECC) computation logic and the overall even parity computation logic. Extended Hamming codes are systematic codes, so this data is also fed through as output on the encoded bit output. After all *k* bits are input, the toggle switches are moved to position B. At this point, data from the Hamming ECC logic is shifted out on the encoded bits bus. Finally, the overall parity bit is shifted out when the output select switch is moved to position C.

In order to encode the product code, each data bit is fed as input both into a row LFSR and a column LFSR. Note that only one row LFSR is necessary for the entire block, since data is written as input in row order. However, each column of the array shall be encoded with a separate LFSR. Each column LFSR is clocked for only one bit of the row, so a more efficient method of column encoding is to store the column LFSR states in a k_x x $(n_y - k_y)$ storage memory. A single LFSR can then be used for all columns of the array. With each bit input, the appropriate column LFSR state is read from the memory, clocked, and written back to the memory.

The encoding process is demonstrated here with an example. Assume a two-dimensional (8,4)x(8,4) extended Hamming product code is to be encoded. This block has 16 data bits, and 64 total encoded bits. Table 97 shows the original 16 data bits denoted by D_{yx}, where *y* corresponds to a column and *x* corresponds to a row.

Table 97—Original data for encoding

D_{11}	D_{21}	D_{31}	D_{41}
D_{12}	D_{22}	D_{32}	D_{42}
D_{13}	D_{23}	D_{33}	D_{43}
D_{14}	D_{24}	D_{34}	D_{44}

The first four bits of the array are fed into the row encoder input in the order D_{11}, D_{21}, D_{31}, D_{41}. Each bit is also fed as input into a unique column encoder. Again, a single column encoder may be used, with the state of each column stored in a memory. After the fourth bit is fed into the input, the first row encoder ECC bits are shifted out.

This process continues for all four rows of data. At this point, 32 bits have been taken as output from the encoder, and the four column encoders are ready to shift out the column ECC bits. This data is shifted out at the end of the row. This continues from the remaining 3 rows of the array. Table 98 shows the final encoded block with the 48 generated ECC bits denoted by E_{yx}.

Table 98—Encoded block

D_{11}	D_{21}	D_{31}	D_{41}	E_{51}	E_{61}	E_{71}	E_{81}
D_{12}	D_{22}	D_{32}	D_{42}	E_{52}	E_{62}	E_{72}	E_{82}
D_{13}	D_{23}	D_{33}	D_{43}	E_{53}	E_{63}	E_{73}	E_{83}
D_{14}	D_{24}	D_{34}	D_{44}	E_{54}	E_{64}	E_{74}	E_{84}
E_{15}	E_{25}	E_{35}	E_{45}	E_{55}	E_{65}	E_{75}	E_{85}
E_{16}	E_{26}	E_{36}	E_{46}	E_{56}	E_{66}	E_{76}	E_{86}
E_{17}	E_{27}	E_{37}	E_{47}	E_{57}	E_{67}	E_{77}	E_{87}
E_{18}	E_{28}	E_{38}	E_{48}	E_{58}	E_{68}	E_{78}	E_{88}

Transmission of the block over the channel occurs in a linear manner; all bits of the first row are transmitted left to right, followed by the second row, etc. This allows for the construction of a near zero-latency encoder, since the data bits can be sent immediately over the channel, with the ECC bits inserted as necessary. For the (8,4)x(8,4) example, the output order for the 64 encoded bits is D_{11}, D_{21}, D_{31}, D_{41}, E_{51}, E_{61}, E_{71}, E_{81}, D_{12}, D_{22}, … ,E_{88}.

For easier readability, the following notation is used:

- The codes defined for the rows (x-axis) are binary (n_x,k_x) block codes.
- The codes defined for the columns (y-axis) are binary (n_y,k_y) block codes.
- Data bits are noted D_{yx} and parity bits are noted E_{yx}.

a) *Shortened BTC:* To match packet sizes, removing symbols from the array shortens a product code. In general, rows or columns are removed until the appropriate size is reached. Codes selected shall have an integral number of information bytes. Different shortening approaches are applicable for BTC. In one method, rows and columns are deleted completely from an initial BTC array. For example, a 253-byte code is generated by starting with (64,57) constituent codes and deleting thirteen rows and eleven columns. Another method uses a more systematic two-dimensional shortening. For example, a 128-byte BTC code is composed of (64,57) constituent codes which are shortened by 25 rows and 25 columns, as described in Figure 119. The end result is a (39,32)x(39,32) array which is capable of encoding 32x32=1024 bits (128 bytes) of data. Table 99 summarizes these example codes. A method for determining codes for payload sizes different than these examples is given at the end of this subclause.

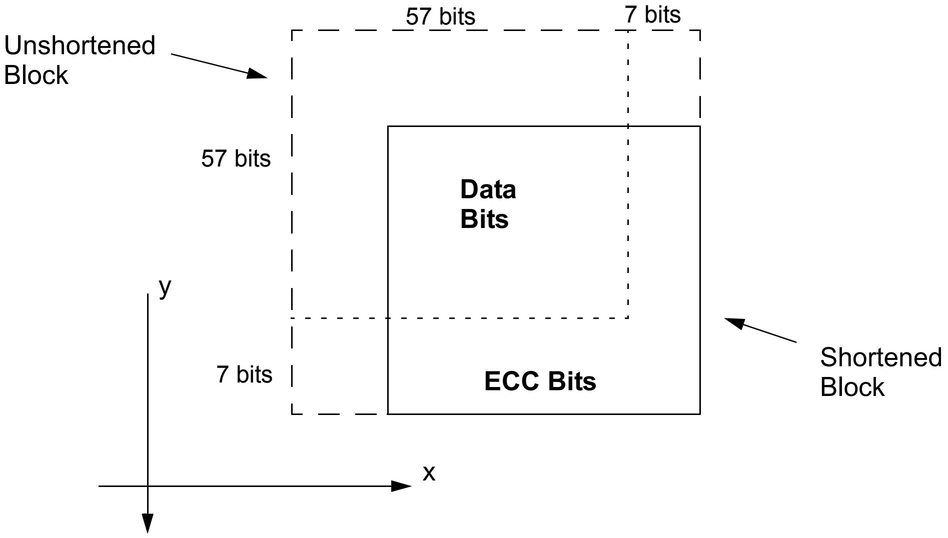


Figure 119—Structure of shortened 2 D block

Modifications to the encoder to support shortening are minimal. Since shortened bits are always zero, and zeros input to the encoder LFSR result in a zero state, the shortened bits can simply be ignored for the purpose of encoding. The encoder simply needs to know how many bits per row to input to the row LFSR before shifting out the result. Similarly, it must know the number of columns to input to the column encoders.

Transmission of the resultant code block shall start with the first data bit in the first row, proceed left to right and then row by row from top to bottom.

Table 99—Required block codes for the BTC option for the downlink channel

Code	(39,32)x(39,32)	(53,46)x(51,44)
Aggregate Code Rate	0.673	0.749
Uplink/Downlink/Both	Downlink	Downlink
Block size (payload bits)	1024 (128 bytes)	3136 (392 bytes)

b) *Interleaving:* When using the Block Turbo Coding, two modes of bit interleaving shall be supported. The interleaver mechanism shall be implemented by writing information bits into the encoder memory and reading out the encoded bits as follows:

1) *Interleaver type 1:* No interleaver. In this mode the encoded bits are read from the encoder row by row, in the order that they were written.

2) *Interleaver type 2:* Block interleaver. In this mode, the encoded bits are read from the encoder after the first *k*2 rows (Figure 117) are written into the encoder memory. The bits are read column by column, proceeding from the top position in the first column.

3) *Interleaver type 3:* Reserved. It is expected that other interleaving methods may yield better performance in some cases. So, this Interleaver type 3 has been reserved for future definition.

c) *Block mapping to the signal constellation:* The first encoded bit out shall be the lsb, which is the first bit written into the encoder.

d) *Method for determining codes for payload size different than the listed examples:* The following text describes a method for performing additional codeword shortening when the input block of data does not match exactly the codeword information size.

1) Take the required payload as specified in bytes and convert it to bits (i.e., multiply by 8).

2) Take the square root of the resultant number.

3) Round the result up to the next highest integer.

4) Select the smallest base constituent code from the available list that has a *k* value equal to or greater than the value determined in step 3.

5) Subtract the value determined in step 3 from the *k* value selected in step 4. This value represents the number of rows and columns that need to be shortened from the base constituent code selected in step 4.

This method will generally result in a code block whose payload is slightly larger than required in step 1 above. In order to address the residual bits, the column dimension (n_y,k_y) should be shortened as needed and, as needed, zero bits may be stuffed into the last bits of the last row of the resulting code matrix. The zero bits in the last row should be discarded at the receiver.

Example: If a 20-byte payload code is desired, a (32,26)x(32,26) code is shortened by 13 rows and by 13 columns, resulting in a (19,13)x(19,13) code. There are 9 bits left over which are stuffed with zeros. Data input to the defined encoder is 160 data bits followed by 9 zero bits. The code block is transmitted starting with the bit in row 1 column 1 (the lsb), then left to right, and then row by row.

8.2.5.4.5 Definition of parameters for burst profile (DIUC=0)

The burst profile with DIUC=0 shall be configured with the parameters in Table 100.

Table 100—Parameters for burst profile (DIUC=0)

Parameter	Value	Comment
Modulation type	1	QPSK
FEC Code Type	2	RS+BCC
RS information bytes (*K*)	26	
RS parity bytes (*R*)	20	
BCC Code Type	1	(24,16)
Last codeword length	1	fixed

8.2.5.4.6 Coding of the control portion of the frame

The frame control section of the downlink frame (as defined in 8.2.5.1) shall be encoded with a fixed set of parameters known to the SS at initialization in order to ensure that all subscriber stations can read the information. The modulation shall be QPSK, and the data shall be encoded with an outer (46,26) Reed–Solomon code and an inner (24,16) convolutional code. There shall be a minimum of 2 codewords per control portion of the frame when a downlink allocation map is present. When an uplink map is present, it shall be concatenated with the Downlink Allocation map to increase efficiency. This operation mode shall be designated as TDM Burst Profile 1 (DIUC = 0). Stuff bytes (FF_{hex}) shall be appended as necessary to the end of the control messages to fill up the minimum number of codewords.

8.2.5.4.7 Downlink modulation

To maximize utilization of the airlink, the PHY uses a multilevel modulation scheme. The modulation constellation can be selected per subscriber based on the quality of the RF channel. If link conditions permit, then a more complex modulation scheme can be utilized to maximizing airlink throughput while still allowing reliable data transfer. If the airlink degrades over time, possibly due to environmental factors, the system can revert to the less complex constellations to allow more reliable data transfer.

In the downlink, the BS shall support QPSK and 16-QAM modulation and, optionally, 64-QAM.

The sequence of modulation bits shall be mapped onto a sequence of modulation symbols $S(k)$, where k is the corresponding symbol number. The number of bits per symbol depends on the modulation type. For QPSK, $n = 2$; for 16-QAM, $n = 4$; and for 64-QAM, $n = 6$. $B(m)$ denotes the modulation bit of a sequence to be transmitted, where m is the bit number (m ranges from 1 through n). In particular, $B(1)$ corresponds to the first bit entering the modulator, $B(2)$ corresponds to the second bit entering the modulation, and so on.

In changing from one burst profile to another, the BS shall use one of two power adjustment rules: maintaining constant constellation peak power (power adjustment rule=0), or maintaining constant constellation mean power (power adjustment rule=1). In the constant peak power scheme, corner points are transmitted at equal power levels regardless of modulation type. In the constant mean power scheme, the signal is transmitted at equal mean power levels regardless of modulation type. The power adjustment rule is configurable through the DCD Channel Encoding parameters (11.1.2.1).

At the end of each burst, the final FEC-encoded message might not end exactly on a PS boundary. If this is the case, the end of the encoded message to the start of the next burst shall be filled with zero bits.

The complex modulation symbol $S(k)$ shall take the value $I + jQ$. The following subsections apply to the base-band part of the transmitter.

Figure 120 and Table 101 describe the bit mapping for QPSK modulation.

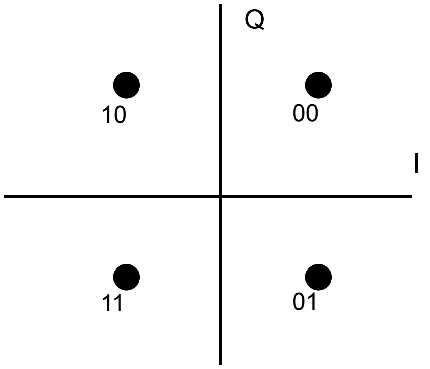


Figure 120—QPSK constellation

Table 101—QPSK bits to symbol mapping

B(1)	B(2)	I	Q
0	0	1	1
0	1	1	–1
1	0	–1	1
1	1	–1	–1

Figure 121 and Table 102 describe the bit mapping for 16-QAM modulation.

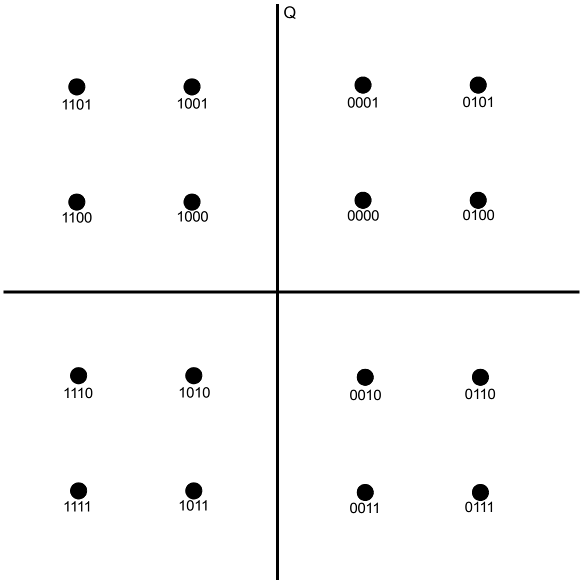


Figure 121— 16-QAM constellation (gray-coded)

Table 102—16–QAM bits to symbol mapping

B(1)	B(2)	B(3)	B(4)	I	Q
0	1	0	1	3	3
0	1	0	0	3	1
0	1	1	0	3	–1
0	1	1	1	3	–3
0	0	0	1	1	3
0	0	0	0	1	1
0	0	1	0	1	–1
0	0	1	1	1	–3
1	0	0	1	–1	3
1	0	0	0	–1	1
1	0	1	0	–1	–1
1	0	1	1	–1	–3
1	1	0	1	–3	3
1	1	0	0	–3	1
1	1	1	0	–3	–1
1	1	1	1	–3	–3

Figure 122 and Table 103 describe the bit mapping for 64-QAM modulation.

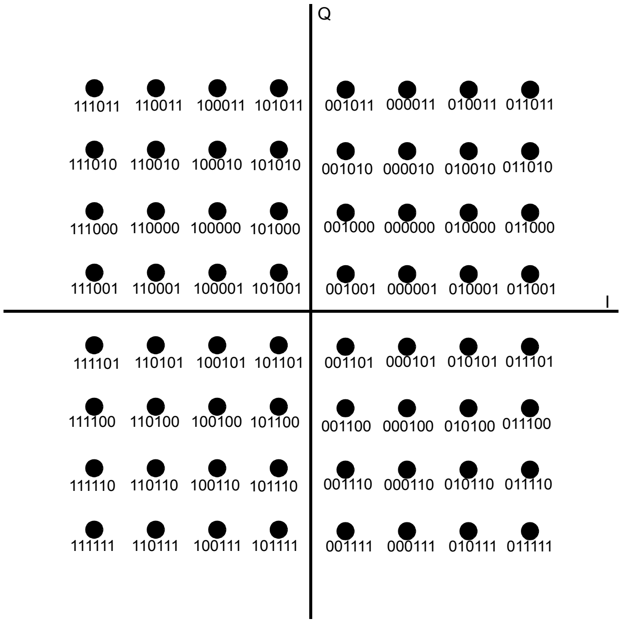


Figure 122— 64-QAM constellation (gray-coded)

Table 103—64-QAM bits to symbol mapping

B(1)	B(2)	B(3)	B(4)	B(5)	B(6)	I	Q
0	1	1	0	1	1	7	7
0	1	1	0	1	0	7	5
0	1	1	0	0	0	7	3
0	1	1	0	0	1	7	1
0	1	1	1	0	1	7	–1
0	1	1	1	0	0	7	–3
0	1	1	1	1	0	7	–5
0	1	1	1	1	1	7	–7
0	1	0	0	1	1	5	7
0	1	0	0	1	0	5	5
0	1	0	0	0	0	5	3
0	1	0	0	0	1	5	1
0	1	0	1	0	1	5	–1
0	1	0	1	0	0	5	–3
0	1	0	1	1	0	5	–5
0	1	0	1	1	1	5	–7
0	0	0	0	1	1	3	7

Table 103—64-QAM bits to symbol mapping *(continued)*

B(1)	B(2)	B(3)	B(4)	B(5)	B(6)	I	Q
0	0	0	0	1	0	3	5
0	0	0	0	0	0	3	3
0	0	0	0	0	1	3	1
0	0	0	1	0	1	3	–1
0	0	0	1	0	0	3	–3
0	0	0	1	1	0	3	–5
0	0	0	1	1	1	3	–7
0	0	1	0	1	1	1	7
0	0	1	0	1	0	1	5
0	0	1	0	0	0	1	3
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	–1
0	0	1	1	0	0	1	–3
0	0	1	1	1	0	1	–5
0	0	1	1	1	1	1	–7
1	0	1	0	1	1	–1	7
1	0	1	0	1	0	–1	5
1	0	1	0	0	0	–1	3
1	0	1	0	0	1	–1	1
1	0	1	1	0	1	–1	–1
1	0	1	1	0	0	–1	–3
1	0	1	1	1	0	–1	–5
1	0	1	1	1	1	–1	–7
1	0	0	0	1	1	–3	7
1	0	0	0	1	0	–3	5
1	0	0	0	0	0	–3	3
1	0	0	0	0	1	–3	1
1	0	0	1	0	1	–3	–1
1	0	0	1	0	0	–3	–3
1	0	0	1	1	0	–3	–5
1	0	0	1	1	1	–3	–7
1	1	0	0	1	1	–5	7
1	1	0	0	1	0	–5	5

Table 103—64-QAM bits to symbol mapping *(continued)*

B(1)	B(2)	B(3)	B(4)	B(5)	B(6)	I	Q
1	1	0	0	0	0	–5	3
1	1	0	0	0	1	–5	1
1	1	0	1	0	1	–5	–1
1	1	0	1	0	0	–5	–3
1	1	0	1	1	0	–5	–5
1	1	0	1	1	1	–5	–7
1	1	1	0	1	1	–7	7
1	1	1	0	1	0	–7	5
1	1	1	0	0	0	–7	3
1	1	1	0	0	1	–7	1
1	1	1	1	0	1	–7	–1
1	1	1	1	0	0	–7	–3
1	1	1	1	1	0	–7	–5
1	1	1	1	1	1	–7	–7

8.2.5.4.8 Baseband pulse shaping

Prior to modulation, the *I* and *Q* signals shall be filtered by square-root raised cosine filters. The excess bandwidth factor α shall be 0.25. The ideal square-root raised cosine filter is defined by the following transfer function *H*:

$$H(f) = 1 \qquad \text{for } |f| < f_N(1-\alpha)$$

$$H(f) = \sqrt{\frac{1}{2} + \frac{1}{2}\sin\left[\frac{\pi}{2f_N}\left(\frac{f_N - |f|}{\alpha}\right)\right]} \qquad \text{for } f_N(1-\alpha) \le |f| \le f_N(1+\alpha)$$

$$H(f) = 0 \qquad \text{for } |f| > f_N(1+\alpha)$$

where $f_N = \frac{1}{2T_S} = \frac{R_S}{2}$ is the Nyquist frequency.

8.2.5.4.9 Transmitted waveform

The transmitted waveform at the antenna port *S*(*t*) shall be:

$$S(t) = I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t)$$

where *I*(*t*) and *Q*(*t*) are the filtered baseband (pulse-shaped) signals of the I_k and Q_k symbols, *k* is the discrete symbol index, and f_c is the carrier frequency.

8.2.5.4.10 Summary of downlink PHY parameters

The downlink PHY parameters are summarized in Table 104.

Table 104—Summary of downlink PHY Parameters

Transmission convergence sublayer	Includes 1 pointer byte
Outer Coding	Reed-Solomon over GF(256) Information byte lengths: 6–255 bytes Error correction capability R = 0-32 (T=0-16) BTC (optional) NOTE—There is no inner code selected in this case.
Randomization	$1 + X^{14} + X^{15}$ Initialization: 100101010000000 at the beginning of each burst
Inner Coding	Selectable from the following options: None (24,16) block convolutional code (9,8) parity check code (optional)
Preamble	Frame Start Preamble (32 symbols) Downlink TDMA Burst Preamble (16 symbols)
Modulation	QPSK (mandatory), 16-QAM (mandatory), 64-QAM (optional)
Spectral shaping	α=0.25

8.2.6 Uplink physical layer

8.2.6.1 Uplink subframe

The structure of the uplink subframe used by the SS to transmit to the BS is shown in Figure 123. Three classes of bursts may be transmitted by the SS during the uplink subframe:

a) Those that are transmitted in contention opportunies reserved for Initial Maintenance.

b) Those that are transmitted in contention opportunies defined by Request Intervals reserved for response to multicast and broadcast polls.

c) Those that are transmitted in intervals defined by Data Grant IEs specifically allocated to individual SSs.

Any of these burst classes may be present in any given frame. They may occur in any order and any quantity (limited by the number of available PSs) within the frame, at the discretion of the BS uplink scheduler as indicated by the UL_MAP in the frame control section (part of the downlink subframe).

The bandwidth allocated for Initial Maintenance and Request contention opportunies may be grouped together and is always used with the uplink burst profiles specified for Initial Maintenance Intervals (UIUC=2) and Request Intervals (UIUC=1), respectively. The remaining transmission slots are grouped by SS. During its scheduled bandwidth, an SS transmits with the burst profile specified by the BS.

SS Transition Gaps separate the transmissions of the various SSs during the uplink subframe. The gap allows for ramping down of the previous burst, followed by a preamble allowing the BS to synchronize to the new SS. The preamble and gap lengths are broadcast periodically in the UCD message.

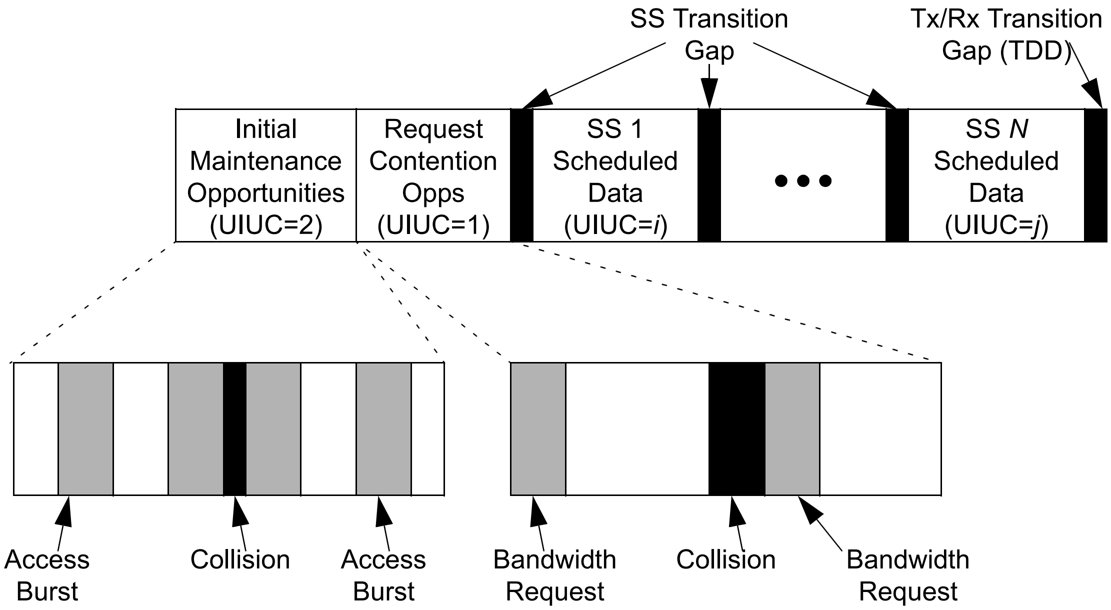

SS Transition Gap
Tx/Rx Transition Gap (TDD)
Initial Maintenance Opportunities (UIUC=2)
Request Contention Opps (UIUC=1)
SS 1 Scheduled Data (UIUC=*i*)
• • •
SS *N* Scheduled Data (UIUC=*j*)
Access Burst
Collision
Access Burst
Bandwidth Request
Collision
Bandwidth Request

Figure 123—Uplink subframe structure

8.2.6.1.1 Uplink burst preamble

Each uplink burst shall begin with an uplink preamble. This preamble is based upon a repetition of a +45 degrees rotated constant amplitude zero auto-correlation (CAZAC) sequence (Milewski [B15]). The preamble length is either 16 symbols or 32 symbols. In the 16-symbol preamble (whose sequence is specified in Table 105), the CAZAC sequence is of length 8 and repeated once. In the 32-symbol preamble (whose sequence is specified in Table 106), the CAZAC sequence is of length 16 and repeated once.

Table 105—16-symbol uplink preamble sequence

Symbol	I	Q	B(1)	B(2)
1 and 9	1	1	0	0
2 and 10	–1	1	1	0
3 and 11	1	1	0	0
4 and 12	1	1	0	0
5 and 13	–1	–1	1	1
6 and 14	–1	1	1	0
7 and 15	–1	–1	1	1
8 and 16	1	1	0	0

Table 106—32-symbol uplink preamble sequence

Symbol	I	Q	B(1)	B(2)
1 and 17	1	1	0	0
2 and 18	1	1	0	0
3 and 19	1	–1	0	1
4 and 20	–1	–1	1	1
5 and 21	–1	–1	1	1
6 and 22	1	1	0	0
7 and 23	1	1	0	0
8 and 24	–1	1	1	0
9 and 25	1	1	0	0
10 and 26	1	1	0	0
11 and 27	–1	1	1	0
12 and 28	1	1	0	0
13 and 29	–1	–1	1	1
14 and 30	1	1	0	0
15 and 31	–1	–1	1	1
16 and 32	1	–1	0	1

The amplitude of the preamble shall depend on the uplink power adjustment rule (8.2.6.3.7). In the case of the constant peak power scheme (power adjustment rule=0), the preamble shall be transmitted such that its constellation points coincide with the outermost constellation points of the modulation scheme in use. In the case of the constant mean power scheme (power adjustment rule=1), it shall be transmitted with the mean power of the constellation points of the modulation scheme in use.

The BS defines the preamble length through the UCD message.

8.2.6.1.2 UL_MAP_Information_Element definition

The format of UL_MAP_Information_Elements shall be as defined in Table 107 and utilized according to 6.2.2.3.4. The Uplink Interval Usage Code shall be one of the values defined in Table 108. The Offset indicates the start time, in units of minislots, of the burst relative to the Allocation Start Time given in the UL-MAP message. The end of the last allocated burst is indicated by allocating a NULL burst (CID = 0 and UIUC = 10) with zero duration. The time instants indicated by the offsets are the transmission times of the first symbol of the burst, including the preamble.

Table 107—UL_MAP_Information_Element

Syntax	Size	Notes
UL_MAP_Information_Element() {		
CID	16 bits	
UIUC	4 bits	
Offset	12 bits	offset, in units of mini-slots, of the preamble relative to the Allocation Start Time
}		

Table 108—Uplink map IE

IE name	UIUC	Connection ID	Description
reserved	0	NA	Reserved for future use.
Request	1	any	Starting offset of request region.
Initial Maintenance	2	broadcast	Starting offset of maintenance region (used in Initial Ranging).
Station Maintenance	3	unicast	Starting offset of maintenance region (used in Periodic Ranging).
Data Grant Burst Type 1	4	unicast	Starting offset of Data Grant Burst Type 1 assignment.
Data Grant Burst Type 2	5	unicast	Starting offset of Data Grant Burst Type 2 assignment.
Data Grant Burst Type 3	6	unicast	Starting offset of Data Grant Burst Type 3 assignment.
Data Grant Burst Type 4	7	unicast	Starting offset of Data Grant Burst Type 4 assignment.
Data Grant Burst Type 5	8	unicast	Starting offset of Data Grant Burst Type 5 assignment.
Data Grant Burst Type 6	9	unicast	Starting offset of Data Grant Burst Type 6 assignment.
Null IE	10	zero	Ending offset of the previous grant. Used to bound the length of the last actual interval allocation.
Empty	11	zero	Used to schedule gaps in transmission.
reserved	12-15	N/A	Reserved.

8.2.6.1.3 Required UCD parameters

The following parameters shall be included in the UCD message:

- Preamble Length

The following parameters may be included in the UCD message and if absent shall have their default values:

- SS Transition Gap
- Roll-off Factor

Uplink Symbol Rate and Frequency are implied by downlink frequency.

8.2.6.1.4 Uplink channel

Since SSs do not transmit in the uplink channel until they have received some minimal configuration information from the BS, it is possible to support several different configurations that can be adjusted on an uplink channel basis or on a burst by burst basis. These parameters, and their ranges, are supported through MAC sublayer signaling, as described in 6.2.2.3.3.

8.2.6.1.5 Uplink_Burst_Profile

Each Uplink_Burst_Profile in the UCD message (6.2.2.3.3) shall include the following parameters:

- Modulation type
- FEC Code Type
- Last codeword length
- Preamble Length
- Scrambler Seed

If the FEC Code Type is 1, 2, or 3 (RS codes), the Uplink_Burst_Profile shall also include

- RS information bytes (*K*)
- RS parity bytes (*R*)

If the FEC Code Type is 2, the Uplink_Burst_Profile shall also include

- BCC code type

If the FEC Code Type is 4, the Uplink_Burst_Profile shall also include

- BTC row code type
- BTC column code type
- BTC interleaving type

Table 109 illustrates the format of the Uplink_Burst_Profile, which is encoded with a Type of 1.

Table 109—Uplink_Burst_Profile format

Syntax	Size	Notes
Type=1	8 bits	
Length	Variable	
reserved	4 bits	shall be set to zero
UIUC	4 bits	
TLV encoded information	Variable	TLV specific

Within each Uplink_Burst_Profile is an unordered list of PHY attributes, encoded as TLV values (see 11.1.1.2).

8.2.6.2 Uplink transmission CS

The uplink transmission CS operation shall be identical to the downlink transmission CS operation, as described in 8.2.5.3.

8.2.6.3 Uplink PHY medium dependent sublayer

The uplink PHY coding and modulation are summarized in the block diagram shown in Figure 124.

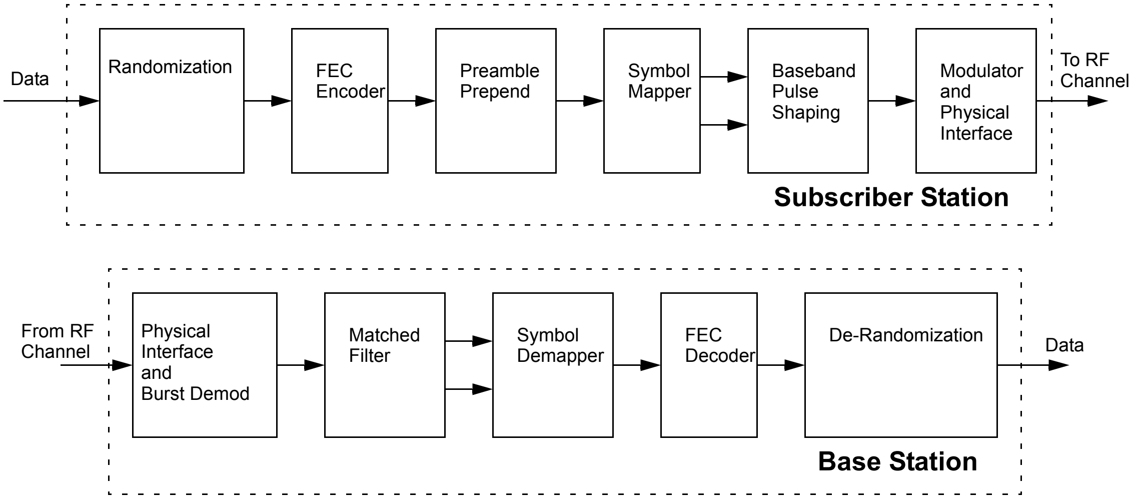


Figure 124—Conceptual block diagram of the uplink PHY

8.2.6.3.1 Randomization for spectrum shaping

The uplink modulator shall implement a randomizer using the polynomial $x^{15}+x^{14}+1$ with the 15-bit programmable Scrambler Seed. At the beginning of each burst, the register is cleared and the Scrambler Seed value is loaded. The Scrambler Seed value shall be used to calculate the scrambler bit, which is combined in an XOR with the first bit of data of each burst [which is the most significant bit (msb) of the first symbol following the last symbol of the preamble].

8.2.6.3.2 Uplink forward error correction

The uplink forward error correction schemes are as described in 8.2.5.4.4, including Table 94.

8.2.6.3.2.1 Outer code for Code Types 1-3, uplink

The Outer Codes for Code Types 1-3 are nearly identical to those of the downlink (8.2.5.4.4.1), with the following exceptions:

a) Fixed Codeword Operation

In the Fixed Codeword Operation, the number of information bytes in each codeword is always the same (K). If the MAC messages in a burst require fewer bytes than are carried by an integral number of Reed-Solomon codewords, stuff bytes (FF_{hex}) shall be added between MAC messages or after the last MAC message so that the total message length is an integral multiple of K bytes.

The SS determines the number of codewords in its uplink burst from the Uplink Map message, which defines the beginning point of each burst, and hence the length. The BS determines the number of codewords in the received uplink burst as it scheduled this transmission event and is aware about its length. Using the burst length, both the SS and the BS calculate the number of full-length RS codewords that can be carried by each burst.

The process used by the SS to encode each burst is identical to the process performed by the BS in Downlink Fixed Codeword Operation (8.2.5.4.4.1).

b) Shortened Last Codeword Operation

In the Shortened Last Codeword Operation, the number of information bytes in the final Reed-Solomon block of each burst is reduced from the normal number *K*, while the number of parity bytes *R* remains the same. The BS tailors the number of information bytes in the last codeword, allowing the SS to transport as many information bytes as possible in each uplink burst. The BS implicitly communicates the number of bytes in the shortened last codeword to the SS via the Uplink Map message, which defines the starting minislot of each burst. The SS uses the Uplink Map information to calculate the number of full-length RS codewords and the length of the shortened last codeword that can be carried within the specified burst size. This calculation shall take into account the number of bytes in the burst used for the preamble and coding bytes as well as the guard time. The BS performs a similar calculation as the SS for its decoding purposes.

To allow the receiving hardware to decode the previous Reed-Solomon codeword, no Reed-Solomon codeword shall have less than 6 information bytes. The number of information bytes carried by the shortened last codeword shall be between 6 and *K* bytes inclusive. In this mode, the BS shall only allocate bursts that result in shortened last codewords of the proper length.

When using Code Type 2, the number of information bytes in the shortened last codeword shall always be an even number so that the total codeword size is also an even number. Both BS and SS shall take this into account when calculating the number of information bytes in the last codeword.

The process used by the SS to encode each burst is identical to the process used by the BS in Downlink Shortened Last Codeword Operation (8.2.5.4.4.1).

8.2.6.3.2.2 Inner code for Code Type 2, uplink

See 8.2.5.4.4.2.

8.2.6.3.2.3 Inner code for Code Type 3, uplink

See 8.2.5.4.4.3.

8.2.6.3.2.4 Code Type 4, uplink

Code Type 4 in the uplink is similar to the downlink case (8.2.5.4.4.4). Some exceptions apply to the uplink due to the smaller payload expected within a burst. For example, using a similar two-dimensional shortening process, a 57-byte code is composed of (32,26) constituent codes which have been shortened by seven rows and two columns as described in Figure 125. The end result is a (30,24)x(25,19) array which is capable of encoding 24x19=456 bits (57 bytes). Table 110 summarizes this code example.

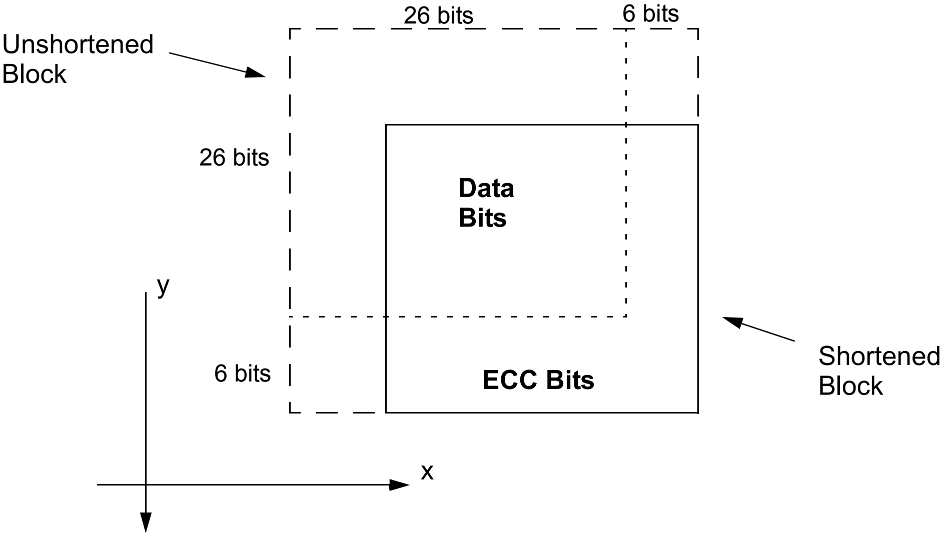


Figure 125—Structure of shortened 2 D block

Table 110—Required block codes for the BTC option for the uplink channel

Code	(30,24)x(25,19)
Aggregate Code Rate	0.608
Uplink/Downlink/Both	Uplink
Block size (payload bits)	456 (57 bytes)

8.2.6.3.3 Shortening of FEC blocks in uplink

Shortening of FEC blocks in the uplink is identical to the handling in the downlink as described in 8.2.5.2 or 8.2.5.4.4.1.

8.2.6.3.4 Number of scheduled uplink bursts per frame

For GPSS SSs, only one scheduled burst (UIUC 4-9) per SS shall be included in the uplink map for any given frame.

8.2.6.3.5 Coding of the Request IE Uplink_Burst_Profile

The uplink burst profile associated with the Request IE (UIUC = 1) shall use Modulation Type = 1 (QPSK) and shall use FEC Code Type = 1 or 2. The other parameters of the Uplink_Burst_Profile encoding shall be chosen such that the resulting uplink burst profile is no less robust than the most robust uplink burst profile associated with any of the Data Grant Burst Type IEs.

8.2.6.3.6 Coding of the Initial Maintenance Uplink_Burst_Profile

The burst profile for the Initial Maintenance UIUC shall be the same as for the frame control section, as defined in 8.2.5.4.6.

8.2.6.3.7 Uplink modulation

The modulation used on the uplink channel shall be variable and set by the BS. QPSK shall be supported, while 16-QAM and 64-QAM are optional, with the mappings of bits to symbols identical to those described in 8.2.5.4.7.

In changing from one burst profile to another, the SS shall use one of two power adjustment rules: maintaining constant constellation peak power (power adjustment rule=0), or maintaining constant constellation mean power (power adjustment rule=1). In the constant peak power scheme, corner points are transmitted at equal power levels regardless of modulation type. In the constant mean power scheme, the signal is transmitted at equal mean power levels regardless of modulation type. The power adjustment rule is configurable through the UCD Channel Encoding parameters (11.1.1.1).

In changing from one modulation scheme to another (i.e., during burst profile change), sufficient RF power amplifier margins should be maintained to prevent violation of emissions masks.

8.2.6.3.8 Baseband pulse shaping

Prior to modulation, the *I* and *Q* signals shall be filtered by square-root raised cosine filters as discussed in 8.2.5.4.8.

8.2.6.3.9 Transmitted waveform

The transmitted waveform shall be as described in 8.2.5.4.9.

8.2.6.3.10 Summary of uplink PHY parameters

Table 111 summarizes the uplink PHY parameters.

Table 111—Summary of uplink PHY parameters

Outer Coding	Reed-Solomon over GF(256) Information byte lengths: 6–255 bytes Error correction capability R = 0–32 (T = 0–16) BTC (optional) NOTE—There is no inner code selected in this case.
Inner Coding	Selectable from the following options: None (24,16) block convolutional code (9,8) parity check code (optional)
Randomization	$x^{15} + x^{14} +1$ Scrambler Seed: 15-bit programmable
Preamble	Based on repetition of 8 symbol or 16 symbol CAZAC sequences
Modulation	QPSK (mandatory);16-QAM (optional); 64-QAM (optional)
Spectral shaping	α = 0.25

8.2.7 Baud rates and channel bandwidths

A large amount of spectrum is potentially available in the 10–66 GHz range for point-to-multipoint systems. Although regulatory requirements vary between different regions, sufficient commonality exists for a default RF channel bandwidth to be specified for each major region. This is necessary in order to ensure that products built to this standard have interoperability over the air interface.

Systems shall use Nyquist square-root raised cosine pulse shaping with a roll-off factor of 0.25 and shall operate on the default RF channel arrangement shown in Table 112. Note that baud rates are chosen to provide an integer number of PSs per frame. The frame duration choice compromises between transport efficiency (with lower frame overhead) and latency.

Table 112—Baud rates and channel sizes for a roll-off factor of 0.25

Channel size (MHz)	Symbol rate (MBaud)	Bit rate (Mbit/s) QPSK	Bit rate (Mbit/s) 16-QAM	Bit rate (Mbit/s) 64-QAM	Recommended Frame Duration (ms)	Number of PSs/frame
20	16	32	64	96	1	4000
25	20	40	80	120	1	5000
28	22.4	44.8	89.6	134.4	1	5600

Due to wide variations in local regulations, no frequency plan is specified in this standard. No single plan can accommodate all cases. For example, the 24.5-26.5 GHz band in Europe is regulated by CEPT requirements concerning specific duplex spacing and rasters. This does not match a similar spectrum allocation in North America.

8.2.8 Radio subsystem control

8.2.8.1 Synchronization technique

The downlink demodulator typically provides an output reference clock that is derived from the downlink symbol clock. This reference can then be used by the subscriber station to provide timing for rate critical interfaces when the downlink clock is locked to an accurate reference at the BS.

Accurate uplink time slot synchronization is supported through a ranging calibration procedure defined by the MAC sublayer to ensure that uplink transmissions by multiple users do not interfere with each other. Therefore, the PHY needs to support accurate timing estimates at the BS, and the flexibility to finely modify the timing at the SS according to the transmitter characteristics specified in 8.2.9.

8.2.8.2 Frequency control

Frequency control is also a critical component of the PHY. Frequency errors, varying with age and temperature, will exist in radio units, particularly so due to the high carrier frequencies. In order to minimize the complexity of the radio-frequency elements at the SS, the uplink and downlink carrier frequencies shall reference each other. Note that there also exists an initial ranging process for frequency and power calibration. After the initial frequency has been calibrated, periodic measurements of the frequency offset value at the BS shall be made by the PHY and sent to the SS via a MAC message, if required.

8.2.8.3 Power control

As with frequency control, a power control algorithm shall be supported for the uplink channel with both an initial calibration and periodic adjustment procedure without loss of data. The BS should be capable of providing accurate power measurements of the received burst signal. This value can then be compared against a reference level, and the resulting error can be fed back to the SS in a calibration message coming from the MAC sublayer. The power control algorithm shall be designed to support power attenuation due to distance loss or power fluctuations at rates of at most 10 dB/second with depths of at least 40 dB. The exact algorithm implementation is vendor-specific. The total power control range consists of both a fixed portion and a portion that is automatically controlled by feedback. The power control algorithm shall take into account the interaction of the RF power amplifier with different burst profiles. For example, when changing from one burst profile to another, margins should be maintained to prevent saturation of the amplifier and to prevent violation of emissions masks.

8.2.9 Minimum performance

This subclause details the minimum performance requirements for proper operation of systems in the frequency range of 24–32 GHz. The values listed in this subclause apply over the operational environmental ranges of the system equipment.

The philosophy taken in this subclause is to guarantee SS interoperability. Hence, the BS is described only in terms of its transmitter (Table 113), while the SS is described in terms of both its transmitter (Table 114) and receiver (Table 115). It is expected that BS manufacturers will use SS transmitter performance coupled with typical deployment characteristics (e.g., cell size, channel loading, near-far users, etc.) to profile their receiver equipment emphasizing specific performance issues as they require.

Table 113—Minimum BS transmitter performance

Tx symbol timing accuracy	Peak-to-peak symbol jitter, referenced to the previous symbol zero crossing, of the transmitted waveform, shall be less than 0.02 of the nominal symbol duration over a 2 s period. The peak-to-peak cumulative phase error, referenced to the first symbol time and with any fixed symbol frequency offset factored out, shall be less than 0.04 of the nominal symbol duration over a 0.1 s period. The Tx symbol timing accuracy shall be within +/– 15 ppm of its nominal value (including aging and temperature variations).
Tx RF frequency/accuracy	10–66 GHz/ +/– 10ppm (including aging and temperature variations)
Spectral mask (out of band/block)	Per relevant local regulation requirements (see 8.2.9.2.2 for more details)
Spurious	Per relevant local regulatory requirements
Maximum Ramp Up/Ramp Down Time	24 symbols (6 PSs)
Modulation accuracy (expressed in EVM, as in 8.2.9.2.3)	12% (QPSK); 6% (16-QAM) (Measured with an Ideal Receiver without Equalizer, all transmitter impairments included), and 10% (QPSK); 3% (16-QAM), 1.5% (64-QAM) (Measured with an Ideal Receiver with an Equalizer, linear distortion removed) Note: Tracking loop bandwidth is assumed to be between 1% to 5% optimized per phase noise characteristics. The tracking loop bandwidth is defined in the following way. A lowpass filter with unity gain at DC and frequency response *H*(*f*), has a tracking loop (noise) bandwidth (B_L), defined as the integral of | *H*(*f*)| squared from 0 to the sampling frequency. The output power of white noise passed through an ideal brick wall filter of bandwidth B_L shall be identical to that of white noise passed through any lowpass filter with the same tracking loop (noise) bandwidth.

Table 114—Minimum SS transmitter performance

Tx Dynamic range	40 dB
Tx RMS Power Level at Maximum Power Level setting for QPSK	At least +15 dBm (measured at antenna port)
Tx power level adjustment steps and accuracy	The SS shall adjust its Tx power level, based on feedback from the BS via MAC messaging, in steps of 0.5 dB in a monotonic fashion. [This required resolution is due to the small gap in sensitivities between different burst profiles (3–4 dB typical).]
Tx symbol timing jitter	Peak-to-peak symbol jitter, referenced to the previous symbol zero crossing, of the transmitted waveform, shall be less than 0.02 of the nominal symbol duration over a 2 s period. The peak-to-peak cumulative phase error, referenced to the first symbol time and with any fixed symbol frequency offset factored out, shall be less than 0.04 of the nominal symbol duration over a 0.1 s period.
Symbol clock	Shall be locked to BS symbol clock.
Tx burst timing accuracy	Shall implement corrections to burst timing in steps of ±0.25 of a symbol with step accuracy of ±0.125 of a symbol.
Tx RF frequency/accuracy	SS frequency locking to BS carrier required.
Spectral Mask (out of band/block)	Per relevant local regulation requirements (see 8.2.9.2.2 for more details).
Maximum Ramp Up/Ramp Down Time	24 symbols (6 Physical slots)
Maximum output noise power spectral density when Tx is not transmitting information	–80 dBm/MHz (measured at antenna port)
Modulation accuracy (expressed in EVM, as in 8.2.9.2.3)	As specified in Table 113.

Table 115—Minimum SS receiver performance

BER performance threshold	For BER = 1 × 10^{-3}: QPSK: –94 + $10\log_{10}$(B) 16-QAM: –87 + $10\log_{10}$(B) 64-QAM: –79 + $10\log_{10}$(B) For BER = 1 × 10^{-6}: QPSK: –90 + $10\log_{10}$(B) 16-QAM: –83 + $10\log_{10}$(B) 64-QAM: –74 + $10\log_{10}$(B) NOTE: Measured uncoded in dBm, where B denotes carrier symbol rate in Mbaud. Propagation models of Type 0, 1, or 2 (Table 116) are used.
Maximum Transition time from Tx to Rx and from Rx to Tx	2 μs (TDD) 20 μs (FDD, half-duplex terminal)
1st Adjacent Channel Interference	At BER 10^{-3}, for 3 dB degradation: C/I = –9 (QPSK), –2 (16-QAM), and +5 (64-QAM) At BER 10^{-3}, for 1 dB degradation: C/I = –5 (QPSK), +2 (16-QAM), and +9 (64-QAM) At BER 10^{-6}, for 3 dB degradation: C/I = –5 (QPSK), +2 (16-QAM), and +9 (64-QAM) At BER 10^{-6}, for 1 dB degradation: C/I = –1 (QPSK), +6 (16-QAM), and +13 (64-QAM) NOTE: Measured uncoded, in dB.
2nd Adjacent Channel Interference	At BER 10^{-3}, for 3 dB degradation: C/I = –34 (QPSK), –27 (16-QAM), and –20 (64-QAM) At BER 10^{-3}, for 1 dB degradation: C/I = –30 (QPSK), –22 (16-QAM), and –16 (64-QAM) At BER 10^{-6}, for 3 dB degradation: C/I = –30 (QPSK), –23 (16-QAM), and –16 (64-QAM) At BER 10^{-6}, for 1 dB degradation: C/I = –26 (QPSK), –20 (16-QAM), and –12 (64-QAM) NOTE: Measured uncoded, in dB.

Note—The interfering source shall be a continuous signal of the same modulation type as the primary signal. The spectral mask of the interfering signal shall depend on local regulatory requirements. For example, where ETSI regulations apply, the 1st and 2nd Adjacent Channel Interference test shall be performed with the interfering signal conforming to the ETSI Type C spectral mask. Where alternative masks are permitted, the interfering signal shall conform to the ETSI Type B spectral mask.

8.2.9.1 Propagation conditions

Line of sight radio propagation conditions between BS and SSs are required, to achieve high quality and availability service. Also, the SSs need highly directional antennas, which minimize the number of multipaths and interference from unexpected sources. The intersymbol interference may occur as a consequence of multipaths.

8.2.9.1.1 Propagation models

In this subclause, the propagation models referred to in this specification are defined. No further BER performance degradation should be expected with all propagation model types.

The channel model is expressed as follows:

$$H(j\omega) = C_1\text{*}\exp(-j\text{*}\omega T_1) + C_2\text{*}\exp(-j\text{*}\omega T_2) + C_3\text{*}\exp(-j\text{*}\omega T_3)$$

Here C_1,C_2, and C_3 are the complex tap amplitudes and T_1,T_2, and T_3 are the tap delays. These parameters are provided in Table 116, where B is the channel baud rate in MBaud and the resulting tap delay is in ns. For example, if B=20 MBaud, then the resulting Type 2 tap delays will be 0, 20, and 40 ns.

Table 116—Propagation models

Propagation model	Tap number i	Tap amplitude C_i	Tap delay T_i
Type 0	1	1.0	0
Type 1	1	0.995	0
	2	0.0995 exp(–j 0.75)	400/B
Type 2	1	0.286 exp(–j 0.75)	0
	2	0.953	400/B
	3	–0.095	800/B

Note: Propagation path parameters are valid for 15 to 25 MBaud.

Type 0 represents a clear line of sight scenario. Type 1 and Type 2 represent typical deployment scenarios with weak multipath components, Type 1 being with better conditions.

8.2.9.1.2 Rain fades

For 10–66 GHz frequencies of operation, the predominant fade mechanism is that resulting from rain attenuation. Fade depths are geographically dependent by rain rate region and are also conditioned by both frequency of operation and link distance. For a given set of equipment transmission parameters and a specified link availability requirement, the rain rate criteria establish the maximum cell radius appropriate to system operation.

An internationally accepted method for computation of rain fade attenuation probability is that defined by ITU-R P.530-8 [B13]. As an example, typical 28 GHz equipment parameters result in a maximum cell radius of about 3.5 km in ITU rain region K. This criteria applies for a link BER = 10^{-6} at a link availability of 99.995%. Further details on this example system model may be found in IEEE Std 802.16.2-2001.

Another important issue is the impact of uncorrelated rain fading between an interference transmission link and a victim transmission link. Under rain fading conditions, the differential rain fading loss between the two transmission paths may have a significant impact on both intrasystem and intersystem link availability. At operational frequencies around 28 GHz, the estimated rain cell diameter is approximately 2.4 km

(ITU-R P.452 [B12]). The effect of rain decorrelation may be estimated based on cell sector size and the specified frequency re-use plan.

A significant mitigation technique for the control of both intrasystem and intersystem interference is the angular discrimination provided by system antennas. The antenna radiation pattern envelope (RPE) discrimination has significance for both clear sky and rain faded propagation conditions. The RPE requirements for aggressive intrasystem frequency reuse plans may exceed the RPE requirements for the control of inter-system coexistence. Recommended antenna RPE characteristics are described in IEEE Std 802.16.2-2001.

8.2.9.2 Transmitter characteristics

Unless stated otherwise, the transmitter requirements are referenced to the transmitter output port and apply with the transmitter tuned to any channel.

8.2.9.2.1 Output power

In the following subclause, power is defined as the time-averaged power when emitting a signal (excluding off-time between bursts), measured over the scrambled bits of one transmitted burst.

The power at which SS or BSs shall operate is specified in the following subclause.

8.2.9.2.1.1 BS

A BS shall not produce an effective isotropic radiated power (EIRP) spectral density exceeding either +14 dBW/MHz or local regulatory requirements. The recommendations in 6.1.1.1 of IEEE Std 802.16.2-2001 should be followed.

8.2.9.2.1.2 SS

An SS shall not produce an EIRP spectral density exceeding either +30 dBW/MHz or local regulatory requirements. The recommendations in 6.1.1.2 of IEEE Std 802.16.2-2001 should be followed.

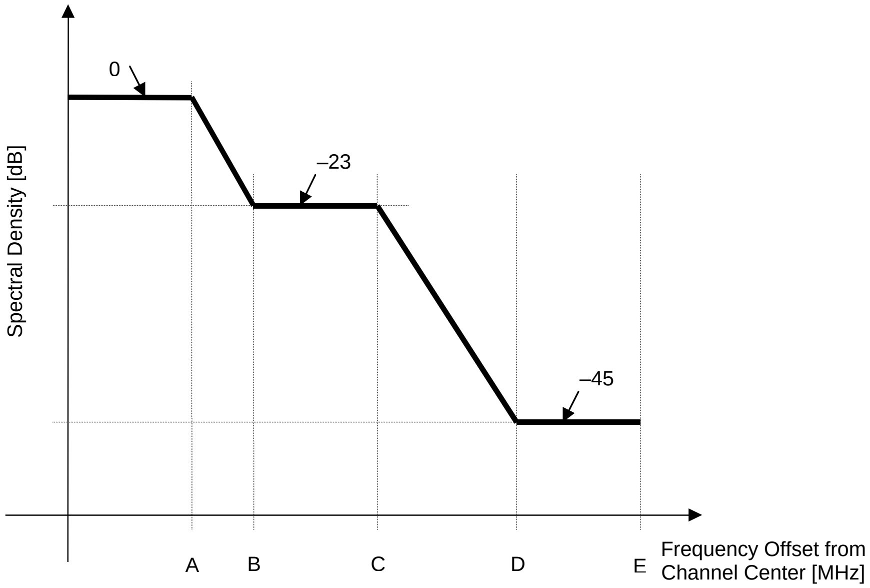
8.2.9.2.2 Emission mask

Local regulation requirements typically dictate emission mask requirements. For example, in its territories, ETSI currently specifies the use of Type A (QPSK), Type B (16-QAM) and Type C (64-QAM) masks (ETSI EN 301 213-3). These masks are presented in Figure 126, Figure 127, and Figure 128.

In the case of mixed modulation systems (e.g., adaptive modulation), ETSI currently specifies the most stringent mask associated with the highest modulation complexity in the adjacent channels.

NOTE—This requirement is under review by ETSI.

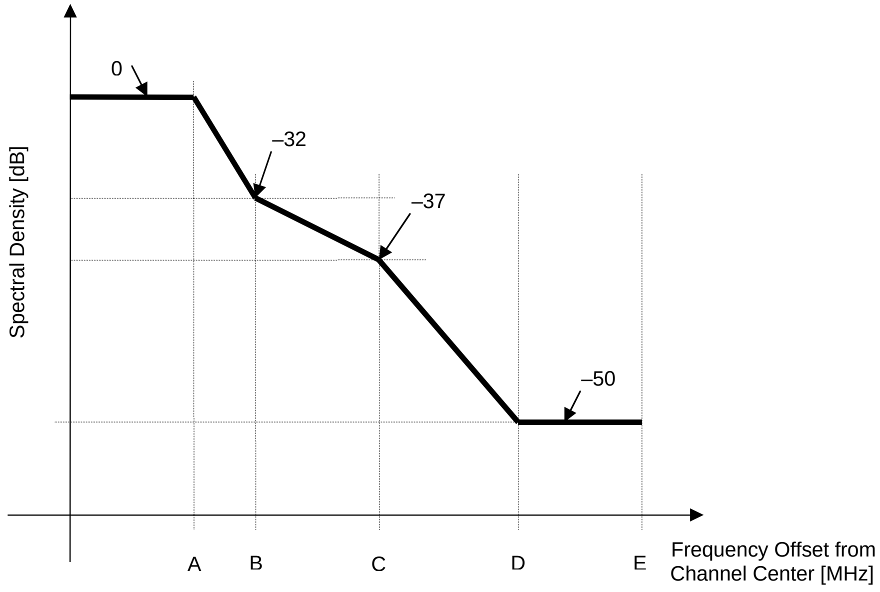
In cases where alternative masks are permitted, the emission mask specified shall be the ETSI Type B mask [ETSI EN 301 213-3] (Figure 127).



Point	A	B	C	D	E
Value in MHz	11c/28	19c/28	25c/28	45c/28	5c/2

Note: c is the channel size in MHz

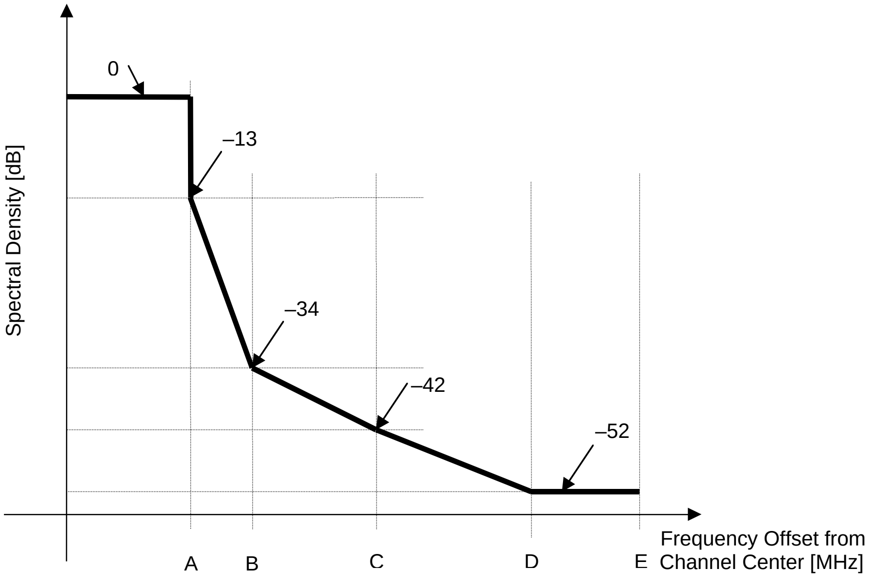
Figure 126—ETSI Type A spectrum mask



Point	A	B	C	D	E
Value in MHz	11.2C/28	22.4C/28	C	2C	5C/2

Note: C is defined as channel size in MHz (i.e., 28 MHz)

Figure 127—ETSI Type B spectrum mask



Point	A	B	C	D	E
Value in MHz	C/2	22.4C/28	C	2C	5C/2

Note: C is defined as channel size in MHz (i.e., 28 MHz)

Figure 128—ETSI Type C spectrum mask

8.2.9.2.3 Modulation accuracy and error vector magnitude (EVM)

The EVM defines the average constellation error with respect to the farmost constellation point power, as illustrated in Figure 129 and defined by the following equation:

$$EVM = \sqrt{\frac{\frac{1}{N}\sum_{1}^{N}(\Delta I^2 + \Delta Q^2)}{S_{\max}^2}}$$

where N is the number of symbols in the measurement period and $S_{\max}$ the maximum constellation amplitude.

The EVM shall be measured over the continuous portion of a burst occupying at least 1/4 of the total transmission frame at maximum power settings.

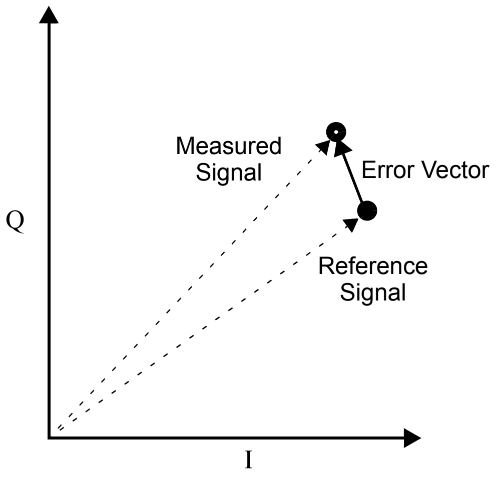


Figure 129—Illustration of EVM

9. Configuration file

9.1 SS IP addressing

9.1.1 Dynamic Host Configuration Protocol (DHCP) fields used by the SS

The following fields shall be present in the DHCP request from the SS and shall be set as described below and encoded as per IETF RFC 2131:

a) The hardware type (htype) shall be set to 1 (Ethernet).

b) The hardware length (hlen) shall be set to 6.

c) The client hardware address (chaddr) shall be set to the 48-bit MAC address associated with the RF interface of the SS.

d) The “client identifier” option shall be included, with the hardware type set to 1, and the value set to the same MAC address as the chaddr field.

e) The “parameter request list” option shall be included. The option codes that shall be included in the list are:

   1) Option code 1 (Subnet Mask)

   2) Option code 2 (Time Offset)

   3) Option code 3 (Router Option)

   4) Option code 4 (Time Server Option)

   5) Option code 7 (Log Server Option)

   6) Option code 60 (Vendor Class Identifier)—A compliant SS shall send the following ASCII coded string in Option code 60: “802.16.”

The following fields are expected in the DHCP response returned to the SS. The SS shall configure itself based on the DHCP response.

a) The IP address to be used by the SS (yiaddr).

b) The IP address of the TFTP server for use in the next phase of the bootstrap process (siaddr).

c) If the DHCP server is on a different network (requiring a relay agent), then the IP address of the relay agent (giaddr). Note: this may differ from the IP address of the first hop router.

d) The name of the SS configuration file to be read from the TFTP server by the SS (file).

e) The subnet mask to be used by the SS (Subnet Mask, option 1).

f) The time offset of the SS from UTC (Time Offset, option 2). This is used by the SS to calculate the local time for use in time-stamping error logs.

g) A list of addresses of one or more routers to be used for forwarding SS-originated IP traffic (Router Option, option 3). The SS is not required to use more than one router IP address for forwarding.

h) A list of time servers (IETF RFC 868) from which the current time may be obtained (Time Server Option, option 4).

i) A list of SYSLOG servers to which logging information may be sent (Log Server Option, option 7).

9.2 SS Configuration File

9.2.1 SS binary configuration file format

The SS-specific configuration data shall be contained in the SS Configuration File which is downloaded to the SS via TFTP. This is a binary file in the same format defined for DHCP vendor extension data (IETF RFC 2132). It shall consist of a number of configuration settings (1 per parameter) each of the form shown in Table 117.

Table 117—Configuration setting format in SS binary configuration file

Type	Length	Value
1 byte	1 byte	1-254 bytes

Here

— Type is a single-byte identifier which defines the parameter;

— Length is a single byte containing the length of the value field in bytes (not including type and length fields);

— Value is from 1 to 254 bytes containing the specific value for the parameter.

The configuration settings shall follow each other directly in the file, which is a stream of bytes (no record markers).

Configuration settings are divided into three types as follows:

— Standard configuration settings which shall be present

— Standard configuration settings which may be present

— Vendor-specific configuration settings

SSs shall be capable of processing all standard configuration settings. SSs shall ignore any configuration setting in the configuration file that it cannot interpret. To allow uniform management of SSs conformant to this specification, conformant SSs shall support a 8192-byte configuration file at a minimum.

Integrity of the configuration file information is provided by the SS MIC (message integrity check). The SS MIC is a digest which ensures that the data sent from the provisioning server were not modified en route. This is not an authenticated digest (it does not include any shared secret).

The file structure is shown in Figure 130:

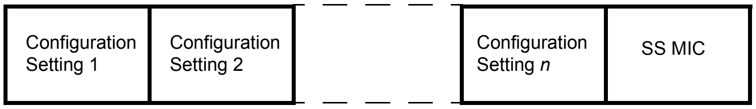


Figure 130—Configuration file structure

9.2.2 Configuration file settings

The following configuration settings shall be included in the configuration file and shall be supported by all SSs:

a) SS MIC Configuration Setting

b) End Configuration Setting (end of data marker).

The following configuration settings may be included in the configuration file and if present shall be supported by all SSs:

a) Software Upgrade Filename Configuration Setting (see 11.3.3)

b) SNMP Write Access Control (multiple) (see 11.3.4)

c) SNMP MIB Object (multiple) (see 11.3.5)

d) Software Server IP Address (see 11.3.6)

e) Pad Configuration Setting (as needed to hit 32-bit boundary in file) (see 11.3.2)

f) Vendor-Specific Configuration Settings

9.2.3 Configuration file creation

The sequence of operations required to create the configuration file is as shown in Figure 131, Figure 132, and Figure 133.

a) Create the type/length/value (TLV) entries for all the parameters required by the SS.

type, length, value for parameter 1
type, length, value for parameter 2
type, length, value for parameter *n*

Figure 131—Create TLV entries for parameters required by the SS

b) Calculate the SS MIC configuration setting as defined in 9.2.3.1 and add to the file following the last parameter using code and length values defined for this field.

type, length, value for parameter 1
type, length, value for parameter 2
type, length, value for parameter *n*
type, length, value for SS MIC

Figure 132—Add SS MIC

c) Add the end of data marker.

type, length, value for parameter 1
type, length, value for parameter 2
type, length, value for parameter *n*
type, length, value for SS MIC
end of data marker

Figure 133—Add end of data marker

9.2.3.1 SS MIC calculation

The SS message integrity check configuration setting shall be calculated by performing a SHA-1 digest over the bytes of the configuration setting fields. It is calculated over the bytes of these settings as they appear in the TFTPed image, without regard to TLV ordering or contents. There are two exceptions to this disregard of the contents of the TFTPed image:

- The bytes of the SS MIC TLV itself are omitted from the calculation. This includes the type, length, and value fields.
- On receipt of a configuration file, the SS shall recompute the digest and compare it to the SS MIC configuration setting in the file. If the digests do not match, then the configuration file shall be discarded.

10. Parameters and constants

10.1 Global values

The BS and SS shall meet the timing requirements contained in Table 118.

Table 118—Parameters and constants

System	Name	Time reference	Minimum value	Default value	Maximum value
BS	DL-MAP Interval	Nominal time between transmission of DL-MAP messages			200 ms
BS	DCD Interval	Time between transmission of DCD messages			10 s
BS	UCD Interval	Time between transmission of UCD messages			10 s
BS	UCD Transition	The time the BS shall wait after repeating a UCD message with an incremented Configuration Change Count before issuing a UL-MAP message referring to Uplink_Burst_Profiles defined in that UCD message	2 ms		
BS	DCD Transition	The time the BS shall wait after repeating a DCD message with an incremented Configuration Change Count before issuing a DL-MAP message referring to Downlink_Burst_Profiles defined in that DCD message	2 ms		
BS	Max MAP Pending	The number of mini-slots that a BS is allowed to map into the future			4096 mini-slot times beyond the Allocation Start Time
BS	Initial Ranging Interval	Time between Initial Maintenance regions assigned by the BS			2 s
BS	CLK-CMP Interval	Time between transmission of CLK-CMP messages	50 ms	50 ms	50 ms
SS	Lost DL-MAP Interval	Time since last received DL-MAP message before downlink synchronization is considered lost			600 ms
SS	Lost UL-MAP Interval	Time since last received UL-MAP message before uplink synchronization is considered lost			600 ms
SS	Contention Ranging Retries	Number of retries on contention Ranging Requests	16		
SS, BS	Invited Ranging Retries	Number of retries on inviting Ranging Requests	16		
SS	Request Retries	Number of retries on bandwidth allocation requests	16		
SS	Registration Request Retries	Number of retries on registration requests	3		

Table 118—Parameters and constants *(continued)*

System	Name	Time reference	Minimum value	Default value	Maximum value
BS	Registration Response Retries	Number of retries on Registration Response	3		
SS	Data Retries	Number of retries on immediate data transmission	16		
BS	SS UL-MAP processing time	Time provided between arrival of the last bit of a UL-MAP at an SS and effectiveness of that map	200 μs		
BS	SS Ranging Response processing time	Time allowed for an SS following receipt of a ranging response before it is expected to reply to an invited ranging request	1 ms		
SS, BS	Minislot size	Size of minislot for uplink transmission. Shall be a power of 2 (in units of PS)	1 PS		
SS, BS	DSx Request Retries	Number of Timeout Retries on DSA/DSC/DSD Requests		3	
SS, BS	DSx Response Retries	Number of Timeout Retries on DSA/DSC/DSD Responses		3	
SS	TFTP Backoff Start	Initial value for TFTP backoff	1 s		
SS	TFTP Backoff End	Last value for TFTP backoff	16 s		
SS	TFTP Request Retries	Number of retries on TFTP request	16		
SS	TFTP Download Retries	Number of retries on entire TFTP downloads	3		
SS	TFTP Wait	The duration between two consecutive TFTP retries	2 min		
SS	Time of Day Retries	Number of Retries per Time of Day Retry Period	3		
SS	Time of Day Retry Period	Time period for Time of Day retries	5 min		
SS	T1	Wait for DCD timeout			5 * DCD interval maximum value
SS	T2	Wait for broadcast ranging timeout			5 * ranging interval
SS	T3	Ranging Response reception timeout following the transmission of a Ranging Request		200 ms	200 ms

Table 118—Parameters and constants *(continued)*

System	Name	Time reference	Minimum value	Default value	Maximum value
SS	T4	Wait for unicast ranging opportunity. If the pending-until-complete field was used earlier by this SS, then the value of that field shall be added to this interval.	30 s		35 s
BS	T5	Wait for Uplink Channel Change response			2 s
SS	T6	Wait for registration response			3 s
SS, BS	T7	Wait for DSA/DSC/DSD Response timeout			1 s
SS, BS	T8	Wait for DSA/DSC Acknowledge timeout			300 ms
BS	T9	Registration Timeout, the time allowed between the BS sending a RNG-RSP (success) to an SS, and receiving a SBC-REQ from that same SS.	300 ms	300 ms	
SS, BS	T10	Wait for Transaction End timeout			3 s
SS	T12	Wait for UCD descriptor			5 * UCD Interval maximum value
BS	T13	The time allowed for an SS, following receipt of a REG-RSP message to send a TFTP-CPLT message to the BS	15 min	15 min	
SS	T14	Wait for DSX-RVD Timeout			200 ms
BS	T15	Wait for MCA-RSP	20 ms	20 ms	
SS	T16	Wait for bandwidth request grant	10 ms		service QoS dependent

10.2 PKM parameter values

Table 119 defines the ranges and default values for the PKM configuration and operational parameters.

Table 119—Operational ranges for privacy configuration settings

System	Name	Description	Minimum value	Default value	Maximum value
BS	AK Lifetime	Lifetime, in seconds, BS assigns to new Authorization Key	1 day (86,400 s)	7 days (604,800 s)	70 days (6,048,000 s)
BS	TEK Lifetime	Lifetime, in seconds, BS assigns to new TEK	30 min (1800 s)	12 h (43,200 s)	7 days (604,800 s)
SS	Authorize Wait Timeout	Auth Req retransmission interval from Auth Wait state	2 s	10 s	30 s
SS	Reauthorize Wait Timeout	Auth Req retransmission interval from Reauth Wait state	2 s	10 s	30 s
SS	Authorization Grace Time	Time prior to Authorization expiration SS begins reauthorization	5 min (300 s)	10 min (600 s)	35 days (3,024,000 s).
SS	Operational Wait Timeout	Key Req retransmission interval from Op Wait state	1 s	1 s	10 s
SS	Rekey Wait Timeout	Key Req retransmission interval from Rekey Wait state	1 s	1 s	10 s
SS	TEK Grace Time	Time prior to TEK expiration SS begins rekeying	5 min (300 s)	1 h (3,600 s)	3.5 days (302,399 s)
SS	Authorize Reject Wait Timeout	Delay before resending Auth Request after receiving Auth Reject	10 s	60 s	10 min (600 s)

For the purposes of protocol testing, it is useful to run the privacy protocol with timer values well below the low end of the operational ranges. The shorter timer values “speed up” privacy’s clock, causing privacy protocol state machine events to occur far more rapidly than they would under an “operational” configuration. While privacy implementations need not be designed to operate efficiently at this accelerated privacy pace, the protocol implementation should operate correctly under these shorter timer values. Table 120 provides a list of shortened parameter values which are likely to be employed in protocol conformance and certification testing.

Table 120—Values for privacy configuration setting for protocol testing

Parameter	Shortened Value
AK Lifetime	5 min (300 s)
TEK Lifetime	3 min (180 s)
Authorization Grace Time	1 min (60 s)
TEK Grace time	1 min (60 s)

The TEK Grace Time shall be less than half the TEK lifetime.

10.3 PHY-specific values

10.3.1 10–66 GHz parameter and constant definitions

10.3.1.1 Physical slot (PS)

Throughout 10.3.1, a PS is the duration of 4 modulation symbols at the symbol rate of the downlink transmission.

10.3.1.2 Symbol rate

The symbol rate shall be in the range 10–32 MBaud, in increments of 100 kBaud.

10.3.1.3 Uplink center frequency

The uplink center frequency shall be a multiple of 250 kHz.

10.3.1.4 Downlink center frequency

The downlink center frequency shall be a multiple of 250 kHz.

10.3.1.5 Tolerated poll jitter

For the 10–66 GHz PHY, the minimum value of the Tolerated Poll Jitter (see 11.4.8.13) shall be 3000 µs.

10.4 Well-known addresses and identifiers

There are several CIDs defined in Table 121 that have specific meaning. These identifiers shall not be used for any other purposes.

Table 121—CIDs

CID	Value	Description
Initial Ranging	0x0000	Used by an SS during initial ranging as part of network entry process.
Basic CID	0x0001—*m*	
Primary Management CIDs	*m*+1—2*m*	
Transport CIDs and Secondary Management CIDs	2*m*+1—0xFeFF	
Multicast Polling CIDs	0xFF00—0xFFFE	An SS may be included in one or more multicast groups for the purposes of obtaining bandwidth via polling. These connections have no associated service flow.
Broadcast CID	0xFFFF	Used for broadcast information that is transmitted on a downlink to all SS.

11. TLV encodings

The following type/length/value (TLV) encodings shall be used in both the configuration file (Clause 9) and MAC Management messages (6.2.2.3).

The format of the length field shall be per the “definite form” of ITU-T X.690. Specifically, if the actual length of the value field is less than or equal to 127 bytes:

- the length of the length field shall be one byte,
- the msb of the length field shall be set to 0, and
- the other 7 bits of the length field shall be used to indicate the actual length of the value field in bytes.

If the length of the value field is more than 127 bytes:

- the length of the length field shall be one byte more than what is actually used to indicate the length of the value field in bytes,
- the most significant bit (msb) of the first byte of the length field shall be set to 1,
- the other 7 bits of the first byte of the length field shall be used to indicate the number of additional bytes of the length field (i.e., excluding the first byte), and
- the remaining bytes (i.e., excluding the first byte) of the length field shall be used to indicate the actual length of the value field.

The following configuration settings shall be supported by all BSs and SSs which are compliant with this specification. MAC management messages that do not contain all required encodings or contain encoding(s) with invalid length(s) shall be silently discarded.

11.1 MAC management message encodings

11.1.1 UCD message encodings

The UCD message encodings are specific to the UCD message (see 6.2.2.3.3).

11.1.1.1 UCD channel encodings

The UCD channel encodings are provided in Table 122.

Table 122—UCD channel encodings

Name	Type (1 byte)	Length	Value (variable-length)
Uplink_Burst_Profile	1		May appear more than once; described below. The length is the number of bytes in the overall object, including embedded TLV items.
Symbol Rate	2	2	Symbol rate, in increments of 10 kBaud.
Frequency	3	4	Uplink center frequency (kHz).
SS Transition Gap	7	1	The time, expressed in PSs, between the end of an SS burst and the beginning of the subsequent burst at the BS. The SS shall take this into account when determining the length of the burst. The SS Transition Gap consumes the last *n* PS of the intervals allocated in the UL-MAP. That is, UL-MAP entries include the time for a burst's ramp down.
Roll-off factor	8	1	0=0.15, 1=0.25, 2=0.35
Power adjustment rule	9	1	0=Preserve Peak Power 1=Preserve Mean Power Describes the power adjustment rule when performing a transition from one burst profile to another.
Contention-based Reservation Time-out	10	1	Number of UL-MAPs to receive before contention-based reservation is attempted again for the same connection.

11.1.1.2 Uplink burst profile encodings for 10–66 GHz systems

The uplink burst profile encodings for 10–66 GHz systems are provided in Table 123.

Table 123—Uplink_Burst_Profile encodings

Name	Type (1 byte)	Length	Value (variable-length)
Modulation type	1	1	1=QPSK, 2=16-QAM, 3=64-QAM
Preamble length	4	1	The number of symbols in the preamble pattern. The preamble consumes the first *n* PS of the intervals allocated in the UL-MAP. That is, UL-MAP entries include the bandwidth for a burst's preamble.
FEC code type	5	1	1 = Reed-Solomon only 2 = Reed-Solomon + Inner (24,16) Block Convolutional Code (BCC) 3 = Reed–Solomon + Inner (9,8) Parity Check Code 4 = Block Turbo Code (Optional) 5–255 = Reserved
RS information bytes (*K*)	6	1	*K*=6–255
RS parity bytes (*R*)	7	1	*R* = 0–32 (error correction capability *T* = 0-16)
BCC code type	8	1	1 = (24,16) 2–255 = Reserved
BTC row code type	9	1	1 = (64,57) Extended Hamming 2 = (32,26) Extended Hamming 3–255 = Reserved.
BTC column code type	10	1	1 = (64,57) Extended Hamming 2 = (32,26) Extended Hamming 3–255 = Reserved
BTC interleaving type	14	1	1 = No interleaver, 2 = Block Interleaving, 3–255 = Reserved.
Scramblerseed	15	2	The 15-bit seed value left-justified in the 2 byte field. Bit 15 is the msb of the first byte, and the lsb of the second byte is not used.
Last codeword length	17	1	1 = fixed; 2 = shortened

11.1.2 DCD message encodings

The DCD message encodings are specific to the DCD message (see 6.2.2.3.1).

11.1.2.1 DCD channel encodings

The DCD Channel Encoding are provided in Table 124.

Table 124—DCD channel encodings

Name	Type (1 byte)	Length	Value (variable-length)
Downlink_Burst_Profile	1		May appear more than once; described below. The length is the number of bytes in the overall object, including embedded TLV items.
BS Transmit Power (average)	2	1	Signed in units of 1 dBm.
Burst FDD/TDD frame duration	4	4	The number of PSs contained in a Burst FDD or TDD frame. Required only for framed downlinks
PHY Type	5	1	The PHY Type to be used.
power adjustment rule	9	1	0=Preserve Peak Power 1=Preserve Mean Power Describes the power adjustment rule when performing a transition from one burst profile to another.

11.1.2.2 Downlink burst profile encodings for 10–66 GHz systems

The downlink burst profile encodings for 10–66 GHz systems are provided in Table 125.

Table 125—Downlink_Burst_Profile encodings

Name	Type (1 byte)	Length	Value (variable-length)
Modulation Type	1	1	1 = QPSK 2 = 16-QAM 3 = 64-QAM
FEC Code Type	2	1	1 = Reed–Solomon only 2 = Reed–Solomon + Inner Block Convolutional Code (BCC) 3 = Reed–Solomon + Inner (9,8) Parity Check Code 4 = Block Turbo Code (Optional) 5–255 = Reserved
RS information bytes (*K*)	3	1	*K*=6–255
RS Parity Bytes (*R*)	4	1	*R* = 0–32 (error correction capability *T* = 0–16)
BCC code type	5	1	1 = (24,16) 2–255 = Reserved
BTC Row code type	6	1	1 = (64,57) Extended Hamming 2 = (32,26) Extended Hamming 3–255 = Reserved
BTC Column code type	7	1	1 = (64,57) Extended Hamming 2 = (32,26) Extended Hamming 3–255 = Reserved
BTC Interleaving type	11	1	1 = No interleaver, 2 = Block Interleaving, 3–255 = Reserved
Last codeword length	12	1	1=fixed; 2=shortened allowed (optional) This allows for the transmitter to shorten the last codeword, based upon the allowable shortened codewords for the particular code type.
DIUC mandatory exit threshold	13	1	C/(N+I) at or below which this DIUC can no longer be used and at which a change to a more robust DIUC is required, in 0.25 dB units. See Figure 59.
DIUC minimum entry threshold	14	1	The minimum C/(N+I) required to start using this DIUC when changing from a more robust DIUC is required, in 0.25 dB units. See Figure 59.
Preamble presence	15	1	0 = burst not preceded with preamble 1 = burst preceded with preamble. If the preamble is present, it consumes the first PSs of the interval.

11.1.3 RNG-REQ message encodings

The encodings in Table 126 are specific to the RNG-REQ message (6.2.2.3.5).

Table 126—RNG-REQ message encodings

Name	Type (1 byte)	Length	Value (variable-length)
Requested Downlink Burst Profile	1	1	DIUC of the downlink burst profile requested by the SS for downlink traffic
SS MAC Address	2	6	The link-layer address of the SS
Ranging Anomalies	3	1	A parameter indicating a potential error condition detected by the SS during the ranging process. Setting the bit associated with a specific condition indicates that the condition exists at the SS. Bit #0 — SS already at maximum power. Bit #1 — SS already at minimum power. Bit #2 — Sum of commanded timing adjustments is too large.

11.1.4 RNG-RSP message encodings

The encodings in Table 127 are specific to the RNG-RSP message (6.2.2.3.6).

Table 127—RNG-RSP message encodings

Name	Type (1 byte)	Length	Value (variable-length)
Timing Adjust	1	4	Tx timing offset adjustment (signed 32-bit, units of 1/4 symbols) The time by which to offset frame transmission so that frames arrive at the expected minislot time at the BS.
Power Level Adjust	2	1	Tx Power offset adjustment (signed 8-bit, 0.25 dB units) Specifies the relative change in transmission power level that the SS is to make in order that transmissions arrive at the BS at the desired power.
Offset Frequency Adjust	3	4	Tx frequency offset adjustment (signed 32-bit, Hz units) Specifies the relative change in transmission frequency that the SS is to make in order to better match the BS. (This is fine-frequency adjustment within a channel, not reassignment to a different channel.)
Ranging Status	4	1	Used to indicate whether uplink Messages are received within acceptable limits by BS. 1 = continue, 2 = abort, 3 = success, 4 = rerange
Downlink frequency override	5	4	Center frequency, in kHz, of new downlink channel on which the SS is to redo initial ranging. If this TLV is used, the Ranging Status value shall be set to 2.
Uplink channel ID override	6	1	The identifier of the uplink channel with which the SS is to redo initial ranging (not used with PHYs without channelized uplinks).
Downlink Operational Burst Profile	7	1	This parameter is sent in response to the RNG-REQ Requested Downlink Burst Profile parameter. It contains the least robust DIUC that may be used by the BS for transmissions to the SS.
SS MAC Address	8	6	SS MAC Address in MAC-48 format
Basic CID	9	2	Basic CID assigned by BS at initial access.
Primary Management CID	11	2	Primary Management CID assigned by BS at initial access.

11.1.5 MCA-REQ and MCA-RSP TLV encodings

The type values used shall be those defined in Table 128. The type and length fields shall each be 1 byte in length.

Table 128—Multicast assignment request message encodings

Name	Type (1 byte)	Length	Value (variable-length)
Multicast CID	1	2	
Assignment	2	1	0x00 = Leave multicast group 0x01 = Join multicast group
reserved	3–255	*n*	Reserved for future use

11.2 PKM message encodings

A summary of the TLV encoding format is shown below. The fields are transmitted from left to right.

Type	Length	Value
1 byte	Variable	Length bytes

Type: The Type field is one byte. Values of the PKM Type field are specified in Table 129. Note that Type values between 0 and 127 are defined within the PKM Specification, while values between 128 and 255 are vendor-assigned Attribute Types.

— A PKM server shall ignore Attributes with an unknown Type.

— A PKM client shall ignore Attributes with an unknown Type.

— PKM client and server (i.e., SS and BS) may log receipt of unknown attribute types.

Length: The Length field indicates the length of this attribute’s Value field, in bytes. The length field *does not include* the Type and Length fields.

Value: The Value field is zero or more bytes and contains information specific to the Attribute. The format and length of the Value field is determined by the Type and Length fields.

— Note that a “string” does not require termination by an ASCII NULL because the Attribute already has a length field.

— The format of the value field is one of the five data types shown in Table 130.

Table 129—PKM Attribute types

Type	PKM Attribute
0–5	*reserved*
6	Display-String
7	AUTH-Key
8	TEK
9	Key-Lifetime
10	Key-Sequence-Number
11	HMAC-Digest
12	SAID
13	TEK-Parameters
14	*reserved*
15	CBC-IV
16	Error-Code
17	CA-Certificate
18	SS-Certificate
19	Security-Capabilities
20	Cryptographic-Suite
21	Cryptographic-Suite-List
22	Version
23	SA-Descriptor
24	SA-Type
25	*reserved*
26	*reserved*
27	PKM Configuration Settings
28-255	*reserved*

Table 130—Attribute value data types

Data type	Structure
string	0 – *n* bytes
uint8	8-bit unsigned integer
uint16	16-bit unsigned integer
uint32	32-bit unsigned integer
compound	collection of Attributes

11.2.1 Display string

Description: This Attribute contains a textual message. It is typically used to explain a failure response and might be logged by the receiver for later retrieval by an SNMP manager. Display strings shall be no longer than 128 bytes. A summary of the Display-String Attribute format is shown below. The fields are transmitted from left to right.

Type	Length	Value (string)
6	≥0 and ≤ 128	A string of characters. There is no requirement that the character string be null terminated; the length field always identifies the end of the string.

11.2.2 AUTH-Key

Description: The Authorization Key (Auth-Key) is a 20-byte quantity, from which a key encryption key (KEK), and two message authentication keys (one for uplink requests, and a second for downlink replies) are derived. This Attribute contains a 128-byte quantity containing the Authorization Key RSA-encrypted with the SS's 1024-bit RSA public key. Details of the RSA encryption procedure are given in 7.5. The ciphertext produced by the RSA algorithm shall be the length of the RSA modulus, i.e., 128 bytes.

Type	Length	Value (string)
7	128	128-byte quantity representing an RSA-encrypted Authorization Key.

11.2.3 TEK

Description: This Attribute contains a quantity that is a TEK key, encrypted with a KEK derived from the Authorization Key.

Type	Length	Value (string)
8	8	Encrypted traffic encryption key.

11.2.4 Key lifetime

Description: This attribute contains the lifetime, in seconds, of an Authorization Key or a TEK. It is a 32-bit unsigned quantity representing the number of remaining seconds for which the associated key shall be valid. Note that this attribute can be used as top level attribute (AK) as well as a subattribute (TEK).

Type	Length	Value (uint32)
9	4	— 32-bit quantity representing key lifetime — A key lifetime of zero indicates that the corresponding Authorization Key or TEK is not valid.

11.2.5 Key-Sequence-Number

Description: This Attribute contains sequence number for a TEK or Authorization Key. The 2 or 4-bit quantity, however, is stored in a single byte, with the high-order 6 or 4 bits set to 0. A summary of the Key-Sequence-Number Attribute format is shown below. Note that this attribute can be used as top level attribute (AK) as well as a subattribute (TEK).

Type	Length	Value (uint8)
10	1	2-bit sequence number (TEK), 4-bit sequence number (AK)

11.2.6 HMAC digest

Description: This Attribute contains a keyed hash used for message authentication. The HMAC algorithm is defined in IETF RFC 2104.

Type	Length	Value (string)
11	20-bytes	A 160-bit (20 byte) keyed SHA hash

11.2.7 SAID

Description: This Attribute contains a 16-bit SAID used by the Privacy Protocol as the security association identifier.

Type	Length	Value (uint16)
12	2	16-bit quantity representing an SAID

11.2.8 TEK parameters

Description: This Attribute is a compound attribute, consisting of a collection of subattributes. These sub-attributes represent all security parameters relevant to a particular generation of an SAID’s TEK. A summary of the TEK-Parameters Attribute format is shown below.

Type	Length	Value (compound)
13	variable	The Compound field contains the sub-Attributes as defined in Table 131

Table 131—TEK-parameters subattributes

Attribute	Contents
TEK	TEK, encrypted with the KEK
Key-Lifetime	TEK Remaining Lifetime
Key-Sequence-Number	TEK Sequence Number
CBC-IV	Cipher Block Chaining (CBC) Initialization Vector

11.2.9 CBC-IV

Description: This Attribute contains a value specifying a Cipher Block Chaining Initialization Vector (CBC-IV). A summary of the CBC-IV attribute format is shown below. The fields are transmitted from left to right.

Type	Length	Value (string)
15	Equal to Block length of cipher	CBC-IV

11.2.10 Error code

Description: This Attribute contains a one-byte error code providing further information about an Authorization Reject, Key Reject, Authorization Invalid, or TEK Invalid. A summary of the Error-Code Attribute format is shown below. Table 132 lists code values for use with this Attribute. The BS may employ the nonzero error codes (1–6) listed below; it may, however, return a code value of zero (0). Error code values other than those defined in Table 132 shall be ignored. Returning a code value of zero sends no additional failure information to the SS; for security reasons, this may be desirable.

Type	Length	Value (uint8)	
16	1	Error-Code	Authorization Reject, Authorization Invalid, Key Reject, TEK Invalid

Table 132—Error-code attribute code values

Error Code	Messages	Description
0	All	No information
1	Auth Reject, Auth Invalid	Unauthorized SS
2	Auth Reject, Key Reject	Unauthorized SAID
3	Auth Invalid	Unsolicited
4	Auth Invalid, TEK Invalid	Invalid Key Sequence Number
5	Auth Invalid	Message (Key Request) authentication failure
6	Auth Reject	Permanent Authorization Failure

Error Code 6 (Permanent Authorization Failure) is used to indicate a number of different error conditions affecting the PKM authorization exchange. These include:

a) an unknown manufacturer; i.e., the BS does not have the CA certificate belonging to the issuer of an SS certificate

b) SS certificate has an invalid signature

c) ASN.1 parsing failure during verification of SS certificate

d) SS certificate is on the “hot list”

e) inconsistencies between certificate data and data in accompanying PKM attributes

f) SS and BS have incompatible security capabilities

The common property of these error conditions is that the failure condition is considered permanent; any reattempts at authorization would continue to result in Authorization Rejects. Details about the cause of a Permanent Authorization Failure may be reported to the SS in an optional Display-String Attribute that may accompany the Error-Code Attribute in Authorization Reject messages. Note that providing this additional detail to the SS should be administratively controlled within the BS. The BS may log these Authorization failures, or even trap them to an SNMP manager.

11.2.11 CA certificate

Description: This Attribute is a string attribute containing an X.509 CA Certificate, as defined in 7.6. A summary of the CA-Certificate Attribute format is shown below. The fields are transmitted from left to right.

Type	Length	Value (string)
17	Variable. Length shall not cause resulting MAC management message to exceed the maximum allowed size.	X.509 CA Certificate (DER-encoded ASN.1)

11.2.12 SS certificate

Description: This Attribute is a string attribute containing an SS's X.509 User Certificate, as defined in 7.6. A summary of the SS-Certificate Attribute format is shown below. The fields are transmitted from left to right.

Type	Length	Value (string)
18	Variable. Length shall not cause resulting MAC management message to exceed the maximum allowed size.	X.509 SS Certificate (DER-encoded ASN.1)

11.2.13 Security capabilities

Description: The Security-Capabilities Attribute is a compound attribute whose subattributes identify the version of PKM an SS supports and the cryptographic suite(s) an SS supports.

Type	Length	Value (compound)
19	variable	The Compound field contains the subattributes as defined in Table 133

Table 133—Security-capabilities subattributes

Attribute	Contents
Cryptographic-Suite-List	list of supported cryptographic suites
Version	version of Privacy supported

11.2.14 Cryptographic suite

Type	Length	Value (uint8,uint8,uint8)
20	3	A 24-bit integer identifying the cryptographic suite properties. The most significant byte, as defined in Table 134, indicates the encryption algorithm and key length. The middle byte, as defined in Table 135 indicates the data authentication algorithm. The least significant byte, as defined in Table 136, indicates the TEK Encryption Algorithm.

Table 134—Data encryption algorithm identifiers

Value	Description
0	No data encryption
1	CBC-Mode, 56-bit DES
2-255	*reserved*

Table 135—Data authentication algorithm identifiers

Value	Description
0	No data authentication
1–255	*reserved*

Table 136—TEK encryption algorithm identifier

Value	Description
0	*reserved*
1	3-DES EDE with 128-bit key
2–255	*reserved*

The allowed cryptographic suites are itemized in Table 137.

Table 137—Allowed cryptographic suites

Value	Description
0x000001	No data encryption, no data authentication & 3-DES,128
0x010001	CBC-Mode 56-bit DES, no data authentication & 3-DES,128
all remaining values	*reserved*

11.2.15 Cryptographic-Suite-List

This parameter contains a list of supported Cryptographic-Suites.

Type	Length	Value (compound)
21	5**n*, where *n* equals number of cryptographic suites listed	A list of Cryptographic Suites

11.2.16 Version

Type	Length	Value (uint8)
22	1	A 1-byte code identifying a version of PKM security as defined in Table 138.

Table 138—Version attribute values

Value	Description
0	*reserved*
1	PKM (Initial standard release)
2–255	*reserved*

11.2.17 SA descriptor

Description: The SA-Descriptor Attribute is a compound attribute whose sub-attributes describe the properties of a Security Association (SA). These properties include the SAID, the SA type, and the cryptographic suite employed within the SA.

Type	Length	Value (compound)
23	variable	The Compound field contains the sub-Attributes shown in Table 139.

Table 139—SA-descriptor subattributes

Attribute	Contents
SAID	Security Association ID
SA-Type	Type of SA
Cryptographic-Suite	Cryptographic suite employed within the SA.

11.2.18 SA type

Description: This attribute identifies the type of SA. Privacy defines three SA types: Primary, Static, Dynamic.

Type	Length	Value (uint8)
24	1	A 1-byte code identifying the value of SA-type as defined in Table 140.

Table 140—SA-type attribute values

Value	Description
0	Primary
1	Static
2	Dynamic
3–127	*reserved*
128–255	Vendor-specific

11.2.19 PKM configuration settings

This field defines the parameters associated with PKM operation. It is composed of a number of encapsulated type/length/value fields.

Type	Length	Value (compound)	Scope
27.X	*n*		Auth Reply

11.2.19.1 Authorize wait timeout

The value of the field specifies retransmission interval, in seconds, of Authorization Request messages from the Authorize Wait state.

Type	Length	Value
27.1	4	Authorize Wait Timeout in seconds

11.2.19.2 Reauthorize wait timeout

The value of the field specifies retransmission interval, in seconds, of Authorization Request messages from Reauthorize Wait state.

Type	Length	Value
27.2	4	Reauthorize Wait Timeout in seconds

11.2.19.3 Authorization grace time

The value of this field specifies the grace period for reauthorization, in seconds.

Type	Length	Value
27.3	4	Authorization Grace Time in seconds

11.2.19.4 Operational wait timeout

The value of this field specifies the retransmission interval, in seconds, of Key Requests from the Operational Wait state.

Type	Length	Value
27.4	4	Operational Wait Timeout in seconds

11.2.19.5 Rekey wait timeout

The value of this field specifies the retransmission interval, in seconds, of Key Requests from the Rekey Wait state.

Type	Length	Value
27.5	4	Rekey Wait Timeout in seconds

11.2.19.6 TEK grace time

The value of this field specifies grace period, in seconds, for rekeying the TEK.

Type	Length	Value
27.6	4	TEK Grace time in seconds

11.2.19.7 Authorize reject wait timeout

The value of this field specifies how long an SS waits (seconds) in the Authorize Reject Wait state after receiving an Authorization Reject.

Type	Length	Value
27.7	4	Authorize Reject Wait Timeout in seconds

11.3 Configuration file encodings

These settings are found in only the configuration file. They shall not be forwarded to the BS in the Registration Request. Type fields and length fields are 1 byte in length.

11.3.1 End-of-data marker

This is a special marker for end of data. It has no length or value fields.

Type
255

11.3.2 Pad configuration setting

This has no length or value fields and is only used following the end of data marker to pad the file to an integral number of 32-bit words.

Type
0

11.3.3 Software upgraded filename

The filename of the software upgrade file for the SS. The filename is a fully qualified directory-path name which is in a format appropriate to the server. There is no requirement that the character string be null terminated; the length field always identifies the end of the string. The file is expected to reside on a TFTP server identified in a configuration setting option defined in 11.4.7.

Type	Length	Value
9	*n*	filename

11.3.4 SNMP write-access control

This object makes it possible to disable SNMP “Set” access to individual Management Information Base (MIB) objects. Each instance of this object controls access to all of the writable MIB objects whose Object ID (OID) prefix matches. This object may be repeated to disable access to any number of MIB objects,

Type	Length	Value
10	*n*	OID prefix plus control flag

where *n* is the size of the ASN.1 Basic Encoding Rules [ISO 8825] encoding of the OID prefix plus one byte for the control flag.

The control flag may take the following values:

0 - allow write-access

1 - disallow write-access

Any OID prefix may be used. The Null OID 0.0 may be used to control access to all MIB objects.

When multiple instances of this object are present and overlap, the longest (most specific) prefix has precedence. Thus, one example might be as follows:

someTable	disallow write-access
someTable.1.3	allow write-access

This example disallows access to all objects in some table except for someTable.1.3.

11.3.5 SNMP MIB Object

This object allows arbitrary SNMP MIB objects to be Set via the TFTP-Registration process,

Type	Length	Value
11	*n*	variable binding

where the value is an SNMP VarBind as defined in IETF RFC 1157. The VarBind is encoded in ASN.1 Basic Encoding Rules, just as it would be if part of an SNMP Set request.

The SS shall treat this object as if it were part of an SNMP Set Request with the following caveats:

a) It shall treat the request as fully authorized (it shall not refuse the request for lack of privilege).

b) SNMP Write-Control provisions (see 11.3.4) do not apply.

c) No SNMP response is generated by the SS.

This object may be repeated with different VarBinds to “Set” a number of MIB objects. All such Sets shall be treated as if simultaneous.

Each VarBind shall be limited to 255 bytes.

11.3.6 Software upgrade TFTP server

This object is the IP address of the TFTP server on which the software upgrade file for the SS resides.

Type	Length	Value
21	4 or 16	IP Address

11.4 Common encodings

In the following subclauses, length is in bytes. Type fields and length fields are 1 byte in length.

11.4.1 SS Capabilities encoding

This value field describes the capabilities of a particular SS, i.e., implementation dependent limits on the particular features or number of features which the SS can support. It is composed from a number of encapsulated type/length/value fields. The encapsulated subtypes define the specific capabilities for the SS in question. Note that the subtype fields defined are only valid within the encapsulated capabilities configuration setting string.

Type	Length	Value
5	*n*	—

11.4.1.1 Uplink CID support

This field shows the number of Uplink CIDs the SS can support. The minimum value is 3; an SS shall support a Basic CID, a Primary Management CID, a Secondary Management CID, and 0 or more Transport CIDs.

Type	Length	Value	Scope
5.8	2	Number of Uplink CIDs the SS can support.	REG-REQ REG-RSP

11.4.1.2 Physical parameters supported

This field defines the parameters associated with the physical capabilities of the SS.

Type	Length	Value
5.12	*n*	—

11.4.1.2.1 10–66 GHz PHY SS demodulator types

This field indicates the different modulation types supported by a 10–66 GHz PHY SS for downlink reception. This field is not used for other PHY specifications. A bit value of 0 indicates “not supported” while 1 indicates “supported.”

Type	Length	Value	Scope
5.12.2	1	bit #0: QPSK bit #1: 16-QAM bit #2: 64-QAM bit #3-7: *reserved*, shall be set to 0	SBC-REQ (see 6.2.2.3.23) SBC-RSP (see 6.2.2.3.24)

11.4.1.2.2 10–66 GHz PHY SS modulator types

This field indicates the different modulation types supported by a 10–66 GHz PHY SS for uplink transmission. This field is not used for other PHY specifications. A bit value of 0 indicates "not supported" while 1 indicates "supported."

Type	Length	Value	Scope
5.12.3	1	bit #0: QPSK bit #1: 16-QAM bit #2: 64-QAM bit #3-7: *reserved*, shall be set to 0	SBC-REQ (see 6.2.2.3.23) SBC-RSP (see 6.2.2.3.24)

11.4.1.2.3 10–66 GHz PHY SS downlink FEC types

This field indicates the different FEC types supported by a 10–66 GHz PHY SS for downlink reception. This field is not used for other PHY specifications. A bit value of 0 indicates "not supported" while 1 indicates "supported."

Type	Length	Value	Scope
5.12.5	1	bit #0: Code Type 1 as in Table 94 bit #1: Code Type 2 as in Table 94 bit #2: Code Type 3 as in Table 94 bit #3: Code Type 4 as in Table 94 bit #4-7: *reserved*, shall be set to 0	SBC-REQ (see 6.2.2.3.23) SBC-RSP (see 6.2.2.3.24)

11.4.1.2.4 10–66 GHz PHY SS uplink FEC types

This field indicates the different FEC types supported by a 10–66 PHY GHz SS for uplink transmission. This field is not used for other physical layer specifications. A bit value of 0 indicates "not supported," while 1 indicates "supported."

Type	Length	Value	Scope
5.12.6	1	bit #0: Code Type 1 as in Table 94 bit #1: Code Type 2 as in Table 94 bit #2: Code Type 3 as in Table 94 bit #3: Code Type 4 as in Table 94 bit #4-7: *reserved*, shall be set to 0	SBC-REQ (see 6.2.2.3.23) SBC-RSP (see 6.2.2.3.24)

11.4.1.3 PKM flow control

This field specifies the maximum number of concurrent PKM transactions that may be outstanding.

Type	Length	Value	Scope
5.16	1	0 indicates no limit 1-255 indicate maximum concurrent transactions default = 0	REG-REQ REG-RSP

11.4.1.4 DSx flow control

This field specifies the maximum number of concurrent DSA, DSC, or DSD transactions that may be outstanding.

Type	Length	Value	Scope
5.17	1	0 indicates no limit 1-255 indicate maximum concurrent transactions default = 0	REG-REQ REG-RSP

11.4.1.5 MCA flow control

This field specifies the maximum number of concurrent MCA transactions that may be outstanding.

Type	Length	Value	Scope
5.18	1	0 indicates no limit 1-255 indicate maximum concurrent transactions default = 0	REG-REQ REG-RSP

11.4.1.6 Bandwidth allocation support

This field indicates properties of the SS that the BS needs to know for bandwidth allocation purposes.

Type	Length	Value	Scope
5.15	1	bit #0=0: Grant per Connection bit #0=1: Grant per SS bit #1=0: Half-duplex bit #1=1: Full-duplex bit #2-7: *reserved*; shall be set to zero	SBC-REQ (see 6.2.2.3.23) SBC-RSP (see 6.2.2.3.24)

11.4.1.7 IP version

This field indicates the version of IP used on the Secondary Management Connection.

Type	Length	Value	Scope
5.9	1	bit #0: 4 (default) bit #1: 6 bit #2-7: *reserved*; shall be set to zero	REG-REQ, REG-RSP

11.4.1.8 MAC CRC support

This field indicates whether or not the SS supports MAC level CRC. A value of 0 indicates no CRC support. A value of 1 indicates that the SS supports MAC sublayer CRC.

Type	Length	Value	Scope
5.19	1	0 = no MAC CRC support 1 = MAC CRC support default = 1	RNG-REQ RNG-RSP

11.4.1.9 Multicast polling group CID support

This field indicates the maximum number of simultaneous Multicast Polling Groups the SS is capable of belonging to.

Type	Length	Value	Scope
5.20	1	0-255 default = 4	RNG-REQ RNG-RSP

11.4.2 SS Message integrity check (MIC) configuration setting

The value field contains the SS MIC code. This is used to detect unauthorized modification or corruption of the configuration file.

Type	Length	Value
6	16	d1 d2....... d16

11.4.3 Vendor ID encoding

The value field contains the vendor identification specified by the 3-byte vendor-specific Organizationally Unique Identifier of the SS or BS MAC address.

The Vendor ID shall be used in a REG-REQ and REG-RSP but shall not be used as a stand-alone configuration file element. It may be used as a subfield of the Vendor Specific Information Field in a configuration file. When used as a sub-field of the Vendor Specific Information field, this identifies the Vendor ID of the SSs which are intended to use this information. A vendor ID used in a Registration Request shall be the Vendor ID of the SS sending the request. A vendor ID used in a Registration Response shall be the Vendor ID of the BS sending the response.

Type	Length	Value	Scope
8	3	v1, v2, v3	REG-REQ (see 6.2.2.3.7)

11.4.4 MAC version encodings

Encodings are as defined for the REG-REQ, including MAC version.

Type	Length	Value	Scope
16	1	Version number of the MAC supported on this channel. The current version is 1.	REG-RSP (see 6.2.2.3.8)

11.4.5 Convergence sublayer capabilites

11.4.5.1 Convergence sublayer support

This parameter indicates which service specific sublayers the SS supports.

Type	Length	Value	Scope
5.20	1	Bit#: 0: ATM 1: Packet, IPv4 2: Packet, IPv6 3: Packet, 802.3 4: Packet, 802.1Q VLAN 5: Packet, IPv4 over 802.3 6: Packet, IPv6 over 802.3 7: Packet, IPv4 over 802.1Q VLAN 8: Packet, IPv6 over 802.1Q VLAN 9-15 *reserved*, shall be set to zero	REG-REQ, REG-RSP

11.4.5.2 Maximum number of classifiers

This is the maximum number of admitted Classifiers that the SS is allowed to have.

Type	Length	Value	Scope
5.21	2	Maximum number of simultaneous admitted classifiers	REG-REQ, REG-RSP

The default value is 0 (no limit).

11.4.5.3 Payload header suppression support

This parameter indicates the level of Payload Header Suppression support.

Type	Length	Value	Scope
5.22	2	Value: 0: no PHS support 1: ATM PHS 2: PacketPHS	REG-REQ, REG-RSP

The default value is 0 (no PHS).

11.4.6 Trivial File Transfer Protocol (TFTP) Server Timestamp

This is the sending time of the configuration file in seconds. The definition of time is as in IETF RFC 868.

Type	Length	Value
19	4	Number of seconds since 00:00 1 January 1900

NOTE—The purpose of this parameter is to prevent replay attacks with old configuration files.

11.4.7 TFTP server provisioned SS address

This parameter is the IP Address of the SS requesting the configuration file.

Type	Length	Value
20	4 or 16	IP Address

NOTE—The purpose of this parameter is to prevent IP spoofing during registration.

11.4.8 Service flow encodings

The following fields define the parameters associated with uplink/downlink scheduling for a service flow. It is somewhat complex in that it is composed from a number of encapsulated TLV fields.

Note that the encapsulated uplink and downlink flow classification configuration setting strings share the same subtype field numbering plan, because many of the subtype fields defined are valid for both types of configuration settings except service flow encodings.

Uplink encodings use the type 24. Downlink encodings use the type 25. Entries of the form [24/25] indicate the encoding can be applied to either an uplink or downlink service flow.

11.4.8.1 Service flow identifier

The Service Flow Identifier (SFID) is used by the BS as the primary reference of a service flow. Only the BS may issue a SFID. It uses this parameterization to issue SFIDs in BS-initiated DSA/DSC-Requests and in its DSA/DSC-Response to SS-initiated DSA/DSC-Requests. The SS specifies the SFID of a service flow using this parameter in a DSC-REQ message.

Type	Length	Value	Scope
[24/25].2	4	1 – 4 294 967 295	DSx-REQ DSx-RSP DSx-ACK

11.4.8.2 Connection identifier

The value of this field specifies the CID assigned by the BS to a service flow with a non-null AdmittedQosParamSet or ActiveQosParamSet. This is used in the bandwidth allocation map to assign uplink bandwidth. This field shall be present in BS-initiated DSA-REQ or DSC-REQ message related to establishing an admitted or active uplink service flow. This field shall also be present in DSA-RSP and DSC-RSP messages related to the successful establishment of an admitted or active uplink service flow.

Even though a service flow has been successfully admitted or activated (i.e., has an assigned CID) the SFID shall be used for subsequent DSx message signalling as it is the primary handle for a service flow. If a service flow is no longer admitted or active (via DSC-REQ) its CID may be reassigned by the BS.

Type	Length	Value	Scope
[24/25].3	2	CID	DSx-REQ DSx-RSP DSx-ACK

11.4.8.3 Service Class Name

The value of this field refers to a predefined BS service configuration to be used for this service flow.

Type	Length	Value	Scope
[24/25].4	2 to 16	Zero-terminated string of ASCII characters.	DSx-REQ DSx-RSP DSx-ACK

NOTE—The length includes the terminating zero.

When the Service Class Name is used in a service flow encoding, it indicates that all the unspecified QoS Parameters of the service flow need to be provided by the BS. It is up to the operator to synchronize the definition of Service Class Names in the BS and in the configuration file.

11.4.8.4 Service Flow Error Parameter Set

This field defines the parameters associated with service flow errors.

Type	Length	Value	Scope
[24/25].5.x	*n*	Compound field	DSx-RSP DSx-ACK

A Service Flow Error Parameter Set is defined by the following individual parameters: Error Code, Errored Parameter, and Error Message.

The Service Flow Error Parameter Set is returned in DSA-RSP and DSC-RSP messages to indicate the recipient's response to a service flow establishment request in a DSA-REQ or DSC-REQ message. The Service Flow Error Parameter Set is returned in DSA-ACK and DSC-ACK messages to indicate the recipient's response to the expansion of a Service Class Name in a corresponding DSA-RSP or DSC-RSP.

On failure, the sender shall include one Service Flow Error Parameter Set for each failed service flow requested in the DSA-REQ or DSC-REQ message. On failure, the sender shall include one Service Flow Error Parameter Set for each failed Service Class Name expansion in the DSA-RSP or DSC-RSP message. Service Flow Error Parameter Set for the failed service flow shall include the Error Code and Errored Parameter and may include an Error Message. If some Service Flow Parameter Sets are rejected but other Service Flow Parameter Sets are accepted, then Service Flow Error Parameters Sets shall be included for only the rejected service flows.

On success of the entire transaction, the RSP or ACK message shall not include a Service Flow Error Parameter Set.

Multiple Service Flow Error Parameter Sets may appear in a DSA-RSP, DSC-RSP, DSA-ACK, or DSC-ACK message, since multiple service flow parameters may be in error. A message with even a single Service Flow Error Parameter Set shall not contain any QoS Parameters.

A Service Flow Error Parameter Set shall not appear in any DSA-REQ or DSC-REQ messages.

NOTE—The entire Service Flow Error Parameter Set encoding shall have a total length of less than 256 octets.

11.4.8.4.1 Errored parameter

The value of this parameter identifies the subtype of a requested service flow parameter in error in a rejected service flow request or Service Class Name expansion response. A Service Flow Error Parameter Set shall have exactly one Errored Parameter TLV within a given service flow encoding.

Type	Length	Value	Scope
[24/25].5.1	1	Service Flow Encoding Subtype in Error	DSx-RSP DSx-ACK

11.4.8.4.2 Error code

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in 11.4.12. A Service Flow Error Parameter Set shall have exactly one Error Code within a given service flow encoding.

Type	Length	Value	Scope
[24/25].5.2	1	Confirmation Code except OK (0)	DSx-RSP DSx-ACK

A value of OK(0) indicates that the service flow request was successful. Since a Service Flow Error Parameter Set only applies to errored parameters, this value shall not be used.

11.4.8.4.3 Error message

This subtype is optional in a Service Flow Error Parameter Set. If present, it indicates a text string to be displayed on the SS console and/or log that further describes a rejected service flow request. A Service Flow Error Parameter Set may have zero or one Error Message subtypes within a given service flow encoding.

Type	Length	Value	Scope
[24/25].5.3	*n*	Zero-terminated string of ASCII characters.	DSx-RSP DSx-ACK

NOTE—The length *n* includes the terminating zero.

11.4.8.5 QoS parameter set type

This parameter shall appear within every service flow encoding. It specifies the proper application of the QoS Parameter Set: to the Provisioned set, the Admitted set, and/or the Active set. When two QoS Parameter Sets are the same, a multibit value of this parameter may be used to apply the QoS parameters to more than one set. A single message may contain multiple QoS parameter sets in separate type 24/25 service flow encodings for the same service flow. This allows specification of the QoS Parameter Sets when their parameters are different. Bit 0 is the lsb of the Value field.

For every service flow that is preprovisioned and for every provisioned service flow added after SS initialization, there shall be a service flow encoding that specifies a ProvisionedQoSParamSet. This service flow encoding, or other service flow encoding(s), may also specify an Admitted and/or Active set.

Type	Length	Value	Scope
[24/25].6	1	Bit 0: Provisioned Set Bit 1: Admitted Set Bit 2: Active Set Bits 3-7: *reserved*	DSx-REQ DSx-RSP DSx-ACK

A BS shall handle a single update to each of the Active and Admitted QoS parameter sets. The ability to process multiple service flow Encodings that specify the same QoS parameter set is not required and is left as a vendor-specific function. If a DSA/DSC contains multiple updates to a single QoS parameter set and the vendor does not support such updates, then the BS shall reply with Confirmation Code 2 (reject-unrecognized-configuration-setting).

Table 141 lists values used in Dynamic Service Messages.

Table 141—Values used In Dynamic Service Messages

Value	Messages
001	Apply to Provisioned set only
011	Apply to Provisioned and Admitted set, and perform admission control
101	Apply to Provisioned and Active sets, perform admission control, and activate this service flow
111	Apply to Provisioned, Admitted, and Active sets; perform admission control and activate this service flow
000	Set Active and Admitted sets to Null
010	Perform admission control and apply to Admitted set
100	Check against Admitted set in separate service flow encoding, perform admission control if needed, activate this service flow, and apply to Active set
110	Perform admission control and activate this service flow, apply parameters to both Admitted and Active sets

11.4.8.6 Traffic priority

The value of this parameter specifies the priority assigned to a service flow. Given two Service flows identical in all QoS parameters besides priority, the higher priority service flow should be given lower delay and higher buffering preference. For otherwise nonidentical service flows, the priority parameter should not take precedence over any conflicting service flow QoS parameter. The specific algorithm for enforcing this parameter is not mandated here.

For uplink service flows, the BS should use this parameter when determining precedence in request service and grant generation, and the SS shall preferentially select contention Request opportunities for Priority Request CIDs based on this priority and its Request/Transmission Policy (see 11.4.8.12).

Type	Length	Value	Scope
[24/25].7	1	0 to 7 — Higher numbers indicate higher priority	DSx-REQ DSx-RSP DSx-ACK

NOTE—The default priority is 0.

11.4.8.7 Maximum sustained traffic rate

This parameter is the rate parameter Rate of a token-bucket-based rate limit for packets. Rate is expressed in bits per second and shall take into account all MAC PDUs of the service flow from the byte following the MAC header HCS to the end of the MAC PDU payload. The number of bytes forwarded-(in bytes) is limited during any time interval T by Max(T), as described in the expression

$$\text{Max}(T) = T * (\text{Rate} / 8) + B, \qquad (1)$$

where the parameter B (in units of bytes) is the Maximum Traffic Burst Configuration Setting (see 11.4.8.8).

NOTES

1—This parameter does not limit the instantaneous rate of the service flow.

2—The specific algorithm for enforcing this parameter is not mandated here. Any implementation which satisfies the above equation is conformant.

3—If this parameter is omitted or set to zero, then there is no explicitly enforced traffic rate maximum. This field specifies only a bound, not a guarantee that this rate is available.

The SS shall defer uplink packets that violate (1) and “rate shape” them to meet the expression, up to a limit as implemented by vendor buffering restrictions, or discard non-conforming packets.

The BS shall enforce expression (1) on all uplink data transmissions. The BS may consider unused grants in calculations involving this parameter. The BS may enforce this limit by any of the following methods: (a) discarding over-limit requests, (b) deferring (through zero-length grants) the grant until it is conforming to the allowed limit, or (c) discarding over-limit data packets. A BS shall report this condition to a policy module. If the BS is policing by discarding either packets or requests, the BS shall allow a margin of error between the SS and BS algorithms.

For a downlink service flow, this parameter is only applicable at the BS. The BS shall enforce expression (1) on all downlink data transmissions. The BS shall not forward downlink packets that violates (1) in any interval T. The BS should “rate shape” the downlink traffic by enqueuing packets arriving in excess of (1), and delay them until the expression can be met.

Type	Length	Value	Scope
[24/25].8	4	Rate (in bits per second)	DSx-REQ DSx-RSP DSx-ACK

11.4.8.8 Maximum traffic burst

The value of this parameter specifies the token bucket size B (in bytes) for this service flow as described in expression (1). This value is calculated from the byte following the MAC header HCS to the end of the MAC PDU payload.

Type	Length	Value	Scope
[24/25].9	4	B (bytes)	DSx-REQ DSx-RSP DSx-ACK

NOTE—The specific algorithm for enforcing this parameter is not mandated here. Any implementation which satisfies the above equation is conformant.

11.4.8.9 Minimum reserved traffic rate

This parameter specifies the minimum rate, in bits per second, reserved for this service flow. The BS should be able to satisfy bandwidth requests for a service flow up to its Minimum Reserved Traffic Rate. If less bandwidth than its Minimum Reserved Traffic Rate is requested for a service flow, the BS may reallocate the excess reserved bandwidth for other purposes. The aggregate Minimum Reserved Traffic Rate of all service flows may exceed the amount of available bandwidth. The value of this parameter is calculated from the byte following the MAC header HCS to the end of the MAC PDU payload. If this parameter is omitted, then it defaults to a value of 0 bits per second (i.e., no bandwidth is reserved for the flow by default).

This field is only applicable at the BS and shall be enforced by the BS.

Type	Length	Value	Scope
[24/25].10	4	Rate (in bits per second)	DSx-REQ DSx-RSP DSx-ACK

NOTE—The specific algorithm for enforcing the value specified in this field is not mandated here.

11.4.8.10 Vendor-specific QoS parameters

This allows vendors to encode vendor-specific QoS parameters. The Vendor ID shall be the first TLV embedded inside Vendor-specific QoS Parameters. If the first TLV inside Vendor-specific QoS Parameters is not a Vendor ID, then the TLV shall be discarded (see 11.4.11).

Type	Length	Value	Scope
[24/25].43	*n*		DSx-REQ DSx-RSP DSx-ACK

11.4.8.11 Service flow scheduling type

The value of this parameter specifies which uplink scheduling service is used for uplink transmission requests and packet transmissions. If this parameter is omitted, then the Best Effort service shall be assumed. This parameter is only applicable at the BS. If defined, this parameter shall be enforced by the BS.

Type	Length	Value	Scope
24.15	1	0: *reserved* 1: for Undefined (BS implementation-dependent[a]) 2: for Best Effort 3: for Non-Real-Time Polling Service 4: for Real-Time Polling Service 5: *reserved* 6: for Unsolicited Grant Service 7: through 255 are reserved for future use	DSx-REQ DSx-RSP DSx-ACK

[a] The specific implementation-dependent scheduling service type could be defined in a message of Type 24.43 (Vendor-specific QoS Parameters).

11.4.8.12 Request/transmission policy

The value of this parameter specifies a variety of uplink request and transmission restrictions, including the capability to further restrict the scheduling service rules outlined in Table 58. Each restriction is enabled by setting its associated bit to 1. If a bit is set to 0, the service flow uses the normal rules for the type of service flow (UGS, etc.). Bit #0 is the lsb of the value field.

Type	Length	Value	Scope
[24/25].16	4	Bit #0 – Service flow shall not use broadcast bandwidth request opportunities. Bit#1-*Reserved*. Bit #2 – The service flow shall not piggyback requests with data. Bit #3 – The service flow shall not fragment data. Bit #4 – The service flow shall not suppress payload headers (convergence sublayer parameter) Bit #5 – The service flow shall not pack multiple SDUs (or fragments) into single MAC PDUs. Bit #6 – The service flow shall not include CRC in the MAC PDU. All other bit positions are reserved.	DSx-REQ DSx-RSP DSx-ACK

11.4.8.13 Tolerated jitter

This parameter defines the Maximum delay variation (jitter) for the connection.

Type	Length	Value	Scope
[24/25].18	4	ms	DSx-REQ DSx-RSP DSx-ACK

11.4.8.14 Maximum latency

The value of this parameter specifies the maximum latency between the reception of a packet by the BS or SS on its network interface and the forwarding of the packet to its RF Interface.

If defined, this parameter represents a service commitment (or admission criteria) at the BS or SS and shall be guaranteed by the BS or SS. A BS or SS does not have to meet this service commitment for service flows that exceed their minimum downlink reserved rate.

Type	Length	Value	Scope
[24/25].14	4	ms	DSx-REQ DSx-RSP DSx-ACK

11.4.8.15 Fixed-length versus variable-length SDU indicator

The value of this parameter specifies whether the SDUs on the service flow are fixed-length or variable-length. The parameter is used only if packing is on for the service flow. The default value is 0, i.e., variable-length SDUs.

Type	Length	Value	Scope
[24/25].24	1	0 = variable-length SDUs 1 = fixed-length SDUs default = 0	DSx-REQ DSx-RSP DSx-ACK

11.4.8.16 SDU size

The value of this parameter specifies the length of the SDU for a fixed-length SDU service flow. This parameter is used only if packing is on and the service flow is indicated as carrying fixed-length SDUs. The default value is 49 bytes, i.e., VC-switched ATM cells with PHS.

Type	Length	Value	Scope
[24/25].25	1	Number of bytes. default = 49	DSx-REQ DSx-RSP DSx-ACK

11.4.8.17 Target SAID

The target SAID parameter indicates the SAID onto which the service flow being set up shall be mapped. This parameter may be used only together with the Service Flow Identifier (11.4.8.1).

Type	Length	Value	Scope
[24/25].26	2	SAID onto which SF is mapped	DSA-REQ (BS initiated) DSA-RSP (SS initiated)

11.4.9 CS specific service flow encodings

11.4.9.1 CS specification

This parameter specifies the CS that the connection being set up shall use.

Type	Length	Value	Scope
[24/25].32	1	0: ATM 1: Packet, IPv4 2: Packet, IPv6 3: Packet, 802.3 4: Packet, 802.1Q VLAN 5: Packet, IPv4 over 802.3 6: Packet, IPv6 over 802.3 7: Packet, IPv4 over 802.1Q VLAN 8: Packet, IPv6 over 802.1Q VLAN 9-255 *reserved*	DSA-REQ

11.4.9.2 CS parameter encoding rules

Each CS defines a set of parameters that are encoded within a subindex under the type values listed below. In the cases of IP over IEEE 802.x, the relevant IP and IEEE 802.x parameters shall be included in the DSx-REQ message.

Type	Convergence sublayer
[24/25].99	ATM
[24/25].100	Packet, IPv4
[24/25].101	Packet, IPv6
[24/25].102	Packet, 802.3
[24/25].103	Packet, 802.1Q VLAN

11.4.9.3 Packet CS encodings for configuration and MAC-layer messaging

The following TLV encodings shall be used in both the configuration file, in SS registration requests and in Dynamic Service Messages. All IEEE 802.3/Ethernet specific TLVs are prefixed to begin with a Type value of [24/25].102. Other Type values are defined for other CSs; refer to 11.4.9.2.

11.4.9.3.1 Configuration-file-specific settings

These settings are found in only the configuration file. They shall NOT be forwarded to the BS in the REG-REQ.

11.4.9.3.2 REG-REQ/REG-RSP specific encodings

These encodings are not found in the configuration file, but are included in the REG-REQ. Some encodings are also used in the REG-RSP.

The SS shall include SS Capabilities Encodings in its REG-REQ. If present in the corresponding REG-REQ, the BS shall include SS Capabilities in the REG-RSP.

11.4.9.3.3 QoS-related encodings

The following TLV encodings shall be used in both the configuration file, registration messages, and Dynamic Service messages to encode parameters for packet classification and scheduling.

The following configuration settings shall be supported by all SSs that are compliant with this specification.

11.4.9.3.4 Dynamic service change action

When received in a Dynamic Service Change (DSC) Request, this indicates the action to be taken with this classifier.

Type	Length	Value
[24/25].100.6	1	0 — DSC Add Classifier 1 — DSC Replace Classifier 2 — DSC Delete Classifier

11.4.9.3.5 Classifier error parameter set

This field defines the parameters associated with Classifier Errors.

Type	Length	Value
[24/25].100.8	*n*	Compound

A Classifier Error Parameter Set is defined by the following individual parameters: Errored Parameter, Error Code, and Error Message.

The Classifier Error Parameter Set is returned in DSA-RSP and DSC-RSP messages to indicate the recipient's response to a Classifier establishment request in a DSA-REQ or DSC-REQ message.

On failure, the sender shall include one Classifier Error Parameter Set for each failed Classifier requested in the DSA-REQ or DSC-REQ message. Classifier Error Parameter Set for the failed Classifier shall include the Error Code and Errored Parameter and may include an Error Message. If some Classifier Sets are rejected but other Classifier Sets are accepted, then Classifier Error Parameter Sets shall be included for only the rejected Classifiers. On success of the entire transaction, the RSP or ACK message shall NOT include a Classifier Error Parameter Set.

Multiple Classifier Error Parameter Sets may appear in a DSA-RSP or DSC-RSP message, since multiple Classifier parameters may be in error. A message with even a single Classifier Error Parameter Set shall NOT contain any other protocol Classifier Encodings (e.g., IP, IEEE Std 802.1D-1998, IEEE Std 802.1Q-1998).

A Classifier Error Parameter Set shall NOT appear in any DSA-REQ or DSC-REQ messages.

11.4.9.3.5.1 Errored parameter

The value of this parameter identifies the subtype of a requested Classifier parameter in error in a rejected Classifier request. A Classifier Error Parameter Set shall have exactly one Errored Parameter TLV within a given Classifier Encoding.

Subtype	Length	Value
[24/25].100.8.1	*n*	Classifier Encoding Subtype in Error

If the length is 1, then the value is the single-level subtype where the error was found; e.g., 7 indicates an invalid Change Action. If the length is 2, then the value is the multilevel subtype where the error was found; e.g., 9-2 indicates an invalid IP Protocol value.

11.4.9.3.5.2 Error code

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in 11.4.12. A Classifier Error Parameter Set shall have exactly one Error Code within a given Classifier Encoding.

Subtype	Length	Value
[24/25].100.8.2	1	Confirmation code except OK (0)

A value of OK(0) indicates that the Classifier request was successful. Since a Classifier Error Parameter Set applies only to errored parameters, this value shall not be used.

11.4.9.3.5.3 Error message

This subtype is optional in a Classifier Error Parameter Set. If present, it indicates a text string to be displayed on the SS console and/or log that further describes a rejected Classifier request. A Classifier Error Parameter Set may have zero or one Error Message subtypes within a given Classifier Encoding.

Subtype	Length	Value
[24/25].100.8.3	*n*	Zero-terminated string of ASCII characters

Note—The length *n* includes the terminating zero.

11.4.9.3.6 Packet classification rule

This compound parameter contains the parameters of the classification rule. All parameters pertaining to a specific classification rule shall be included in the same Packet Classification Rule compound parameter.

Type	Length	Value
[24/25].100.9	*n*	Compound

11.4.9.3.6.1 Classifier rule priority

The value of the field specifies the priority for the Classifier, which is used for determining the order of the Classifier. A higher value indicates higher priority.

Classifiers may have priorities in the range 0–255 with the default value being 0.

Type	Length	Value
[24/25].100.9.1	1	0–255

11.4.9.3.6.2 IP Type of Service/DSCP range and mask

The values of the field specify the matching parameters for the IP ToS/DSCP [IETF RFC 2474] byte range and mask. An IP packet with IP ToS byte value “ip-tos” matches this parameter if tos-low <= (ip-tos AND tos-mask) <= tos-high. If this field is omitted, then comparison of the IP packet ToS byte for this entry is irrelevant.

Type	Length	Value
[24/25].100.9.2	3	tos-low, tos-high, tos-mask

11.4.9.3.6.3 Protocol

The value of the field specifies a list of matching values for the IP Protocol field. For IPv6 (IETF RFC 2460), this refers to next header entry in the last header of the IP header chain. The encoding of the value field is that defined by the IANA document "Protocol Numbers." If this parameter is omitted, then comparison of the IP header Protocol field for this entry is irrelevant.

Type	Length	Value
[24/25].100.9.3	*n*	prot1, prot2,...prot n

11.4.9.3.6.4 IP masked source address

This parameter specifies a list of of IP source addresses (designated "src_i") and their corresponding address masks (designated "$smask_i$"). An IP packet with IP source address "ip-src" matches this parameter if src_i = (ip-src AND $smask_i$) for any *i* from 1 to *n*. If this parameter is omitted, then comparison of the IP packet source address for this entry is irrelevant.

Type	Length	Value
[24/25].100.9.4	*n**8 (IPv4) or *n**32 (IPv6)	src_1, $smask_1$,..., src_i, $smask_{i,}$..., src_n, $smask_n$

11.4.9.3.6.5 IP destination address

This parameter specifies a list of IP destination addresses (designated "dst_i") and their corresponding address masks (designated "$dmask_i$"). An IP packet with IP destination address "ip-dst" matches this parameter if dst_i = (ip-dst AND $dmask_i$) for any *i* from 1 to *n*. If this parameter is omitted, then comparison of the IP packet destination address for this entry is irrelevant.

Type	Length	Value
[24/25].100.9.5	*n**8 (IPv4) or *n**32 (IPv6	dst_1, $dmask_1$,..., dst_i, $dmask_{i,}$..., dst_n, $dmask_n$

11.4.9.3.6.6 Protocol source port range

The value of the field specifies a list of nonoverlapping ranges of protocol source port values. Classifier rules with port numbers are protocol specific; i.e., a rule on port numbers without a protocol specification shall not be defined. An IP packet with protocol port value "src-port" matches this parameter if sportlow <= src-port <= sporthigh. If this parameter is omitted, the protocol source port is irrelevant. This parameter is irrelevant for protocols without port numbers.

Type	Length	Value
[24/25].100.9.6	*n**4	sportlow 1, sporthigh 2,...,sportlow *n*, sporthigh *n*

11.4.9.3.6.7 Protocol destination port range

The value of the field specifies a list of non-overlapping ranges of protocol destination port values. Classifier rules with port numbers are protocol specific; i.e., a rule on port numbers without a protocol specification shall not be defined. An IP packet with protocol port value “dst-port” matches this parameter if dportlow <= dst-port <=dporthigh. If this parameter is omitted the protocol destination port is irrelevant. This parameter is irrelevant for protocols without port numbers.

Type	Length	Value
[24/25].100.9.7	*n**4	dportlow 1, dporthigh 2,...,dportlow *n*, dporthigh *n*

11.4.9.3.6.8 Ethernet destination MAC address

This parameter specifies a list of MAC destination addresses (designated “dst_i”) and their corresponding address masks (designated “msk_i”). An Ethernet packet with MAC destination address “etherdst” corresponds to this parameter if dst_i = (etherdst AND msk_i) for any *i* from 1 to *n*. If this parameter is omitted, then comparison of the Ethernet destination MAC address for this entry is irrelevant.

Type	Length	Value
[24/25].100.9.8	*n**12	dst_1, msk_1,..., dst_i, $\text{msk}_{i,}$..., dst_n, msk_n

11.4.9.3.6.9 Ethernet source MAC address

This parameter specifies a list of MAC source addresses (designated “src_i”) and their corresponding address masks (designated “msk_i”). An Ethernet packet with MAC source address “ethersrc” corresponds to this parameter if src_i = (ethersrc AND msk_i) for any *i* from 1 to *n*. If this parameter is omitted, then comparison of the Ethernet source MAC address for this entry is irrelevant.

Type	Length	Value
[24/25].100.9.9	*n**12	src_1, msk_1,..., src_i, $\text{msk}_{i,}$..., src_n, msk_n

11.4.9.3.6.10 Ethertype/IEEE Std 802.2-1998 SAP

The format of the Layer 3 protocol ID in the Ethernet packet is indicated by type, eprot1, and eprot2 as follows:

If type = 0, the rule does not use the Layer 3 protocol type as a matching criteria. If type = 0, eprot1, eprot2 are ignored when considering whether a packet matches the current rule.

If type = 1, the rule applies only to SDUs which contain an Ethertype value. Ethertype values are contained in packets using the DEC-Intel-Xerox (DIX) encapsulation or the Sub-Network Access Protocol (SNAP) encapsulation (IEEE Std 802.2-1998, IETF RFC 1042) format. If type = 1, then eprot1, eprot2 gives the 16-bit value of the Ethertype that the packet shall match in order to match the rule.

If type = 2, the rule applies only to SDUs using the IEEE Std 802.2-1998 encapsulation format with a Destination Service (DSAP) other than 0xAA (which is reserved for SNAP). If type = 2, the lower 8 bits of the eprot1, eprot2 shall match the DSAP byte of the packet in order to match the rule.

If the Ethernet SDU contains an IEEE Std 802.1D-1998 and IEEE Std 802.1Q-1998 Tag header (i.e., Ethertype 0x8100), this object applies to the embedded Ethertype field within the IEEE Std 802.1D-1998 and IEEE Std 802.1Q-1998 header.

Other values of type are reserved. If this TLV is omitted, then comparison of either the Ethertype or IEEE Std 802.2-1998 DSAP for this rule is irrelevant.

Type	Length	Value
[24/25].100.9.10	3	type, eprot1, eprot2

11.4.9.3.6.11 IEEE Std 802.1D-1998 User_Priority

The values of this field specify the matching parameters for the IEEE Std 802.1802.1D-1998 user_priority bits. An Ethernet packet with IEEE Std 802.1D-1998 user_priority value “priority” matches these parameters if pri-low <= priority <= pri-high. If this field is omitted, then comparison of the IEEE Std 802.1D-1998 user_priority bits for this entry is irrelevant.

If this parameter is specified for an entry, then Ethernet packets without IEEE Std 802.1Q-1998 encapsulation shall NOT match this entry. If this parameter is specified for an entry on an SS that does not support forwarding of IEEE Std 802.1Q-1998 encapsulated traffic, then this entry shall NOT be used for any traffic.

Type	Length	Value
[24/25].100.9.11	2	pri-low, pri-high Valid Range: 0–7 for pri-low and pri-high

11.4.9.3.6.12 IEEE Std 802.1Q-1998 VLAN_ID

The value of the field specifies the matching value for the IEEE Std 802.1Q-1998 vlan_id bits. Only the first (i.e. left-most) 12 bits of the specified vlan_id field are significant; the final four bits shall be ignored for comparison. If this field is omitted, then comparison of the IEEE Std 802.1Q-1998 vlan_id bits for this entry is irrelevant.

If this parameter is specified for an entry, then Ethernet packets without IEEE Std 802.1Q-1998 encapsulation shall NOT match this entry. If this parameter is specified for an entry on an SS that does not support forwarding of IEEE Std 802.1Q-1998 encapsulated traffic, then this entry shall NOT be used for any traffic.

Type	Length	Value
[24/25].100.9.12	2	vlan_id1, vlan_id2

11.4.9.3.6.13 Associated payload header suppression index

The Associated Payload Suppression Header Index (PHSI) has a value between 1 and 255 which shall mirror the PHSI value of a payload header suppression rule. Packets matching the Packet Classification Rule containing the Associated PHSI parameter shall undergo PHS according to the corresponding PHS rule.

Type	Length	Value
[24/25].100.9.13	1	index value

11.4.9.3.6.14 Vendor specific classifier parameters

This allows vendors to encode vendor-specific Classifier parameters. The Vendor ID shall be the first TLV embedded inside Vendor Specific Classifier Parameters. If the first TLV inside Vendor Specific Classifier Parameters is not a Vendor ID, then the TLV shall be discarded (see 11.4.11).

Type	Length	Value
[24/25].100.9.255	*n*	

11.4.9.3.6.15 DSC Action

When received in a DSC Request, this indicates the action that shall be taken with this PHS byte string.

Type	Length	Value
[24/25].100.10	1	0 — Add PHS Rule 1 — Set PHS Rule 2 — Delete PHS Rule 3 — Delete all PHS Rules

The “Set PHS Rule” command is used to add the specific TLVs for an undefined payload header suppression rule. It shall NOT be used to modify existing TLVs.

When deleting all PHS Rules any corresponding PHSI shall be ignored.

An attempt to add a PHS Rule which already exists is an error condition.

11.4.9.3.6.16 PHS error parameter set

This field defines the parameters associated with PHS errors.

Type	Length	Value
[24/25].100.11	*n*	compound field

A PHS Error Parameter Set is defined by the following individual parameters: Errored Parameter, Error Code, and Error Message.

The PHS Error Parameter Set is returned in DSA-RSP and DSC-RSP messages to indicate the recipient’s response to a PHS Rule establishment request in a DSA-REQ or DSC-REQ message.

On failure, the sender shall include one PHS Error Parameter Set for each failed PHS Rule requested in the DSA-REQ or DSC-REQ message. PHS Error Parameter Set for the failed PHS Rule shall include the Error Code and Errored Parameter and may include an Error Message. If some PHS Rule Sets are rejected but other PHS Rule Sets are accepted, then PHS Error Parameter Sets shall be included for only the rejected PHS Rules. On success of the entire transaction, the RSP or ACK message shall NOT include a PHS Error Parameter Set.

Multiple PHS Error Parameter Sets may appear in a DSA-RSP or DSC-RSP message, since multiple PHS parameters may be in error. A message with even a single PHS Error Parameter Set shall NOT contain any other protocol PHS Encodings (e.g., IP or IEEE Std 802.1D-1998/IEEE Std 802.1Q-1998).

A PHS Error Parameter Set shall NOT appear in any DSA-REQ or DSC-REQ messages.

11.4.9.3.6.17 Errored parameter

The value of this parameter identifies the subtype of a requested PHS Parameter in error in a rejected PHS request. A PHS Error Parameter Set shall have exactly one Errored Parameter TLV within a given PHS Encoding.

Type	Length	Value
[24/25].100.11.1	1	Payload Header Suppression Encoding Subtype in Error

11.4.9.3.6.18 Error code

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in 11.4.12. A PHS Error Parameter Set shall have exactly one Error Code within a given PHS Encoding.

Type	Length	Value
[24/25].100.11.2	1	Confirmation code except OK(0)

A value of OK(0) indicates that the PHS request was successful. Since a PHS Error Parameter Set only applies to errored parameters, this value shall not be used.

11.4.9.3.6.19 Error message

This subtype is optional in a PHS Error Parameter Set. If present, it indicates a text string to be displayed on the SS console and/or log that further describes a rejected PHS request. A PHS Error Parameter Set may have zero or one Error Message subtypes within a given PHS Encoding.

Type	Length	Value
[24/25].100.11.3	*n*	Zero-terminated string of ASCII characters

The length *n* includes the terminating zero.

11.4.9.3.7 PHS Rule

This field defines the parameters associated with a PHS Rule.

Type	Length	Value
[24/25].100.12	*n*	

11.4.9.3.7.1 Payload Header Suppression Index

The PHSI has a value between 1 and 255 which uniquely references the suppressed byte string. The Index is unique per service flow. The uplink and downlink PHSI values are independent of each other.

Type	Length	Value
[24/25].100.12.1	1	index value

11.4.9.3.7.2 Payload Header Suppression Field (PHSF)

The PHSF is a string of bytes containing the header information to be suppressed by the sending CS and reconstructed by the receiving CS. The MSB of the string corresponds to first byte of the CS-SDU.

Type	Length	Value
[24/25].100.12.2	*n*	string of bytes suppressed

The length *n* shall always be the same as the value for PHSS.

11.4.9.3.7.3 Payload Header Suppression Mask (PHSM)

The value of this field is used to interpret the values in the PHSF. It is used at both the sending and receiving entities on the link. The PHSM allows fields, such as sequence numbers or checksums, which vary in value to be excluded from suppression with the constant bytes around them suppressed.

Type	Length	Value
[24/25].100.12.3	*n*	bit 0: 0 = don't suppress first byte of the suppression field 1 = suppress first byte of the suppression field bit 1: 0 = don't suppress second byte of the suppression field 1 = suppress second byte of the suppression field bit *x*: 0 = don't suppress (*x*+1) byte of the suppression field 1 = suppress (*x*+1) byte of the suppression field

The length I is ceiling(PHSS/8). Bit 0 is the msb of the Value field. The value of each sequential bit in the PHSM is an attribute for the corresponding sequential byte in the PHSF.

If the bit value is a "1," the sending entity should suppress the byte, and the receiving entity should restore the byte from its cached PHSF. If the bit value is a "0," the sending entity should not suppress the byte, and the receiving entity should restore the byte by using the next byte in the packet.

If this TLV is not included, the default is to suppress all bytes.

11.4.9.3.7.4 Payload Header Suppression Size (PHSS)

The value of this field is the total number of bytes in the header to be suppressed and then restored in a service flow that uses PHS.

Type	Length	Value
[24/25].100.12.4	1	number of bytes in the suppression string

This TLV is used when a service flow is being created. For all packets which get classified and assigned to a service flow with PHS enabled, suppression shall be performed over the specified number of bytes as indicated by the PHSS and according to the PHSM. If this TLV is not included in a service flow definition, or is included with a value of 0 bytes, then PHS is disabled. A non-zero value indicates PHS is enabled.

11.4.9.3.7.5 Payload Header Suppression Verification (PHSV)

The value of this field indicates to the sending entity whether or not the packet header contents are to be verified prior to performing suppression. If PHSV is enabled, the sender shall compare the bytes in the packet header with the bytes in the PHSF that are to be suppressed as indicated by the PHSM.

Type	Length	Value
[24/25].100.12.5	1	0 = verify 1 = don't verify

If this TLV is not included, the default is to verify. Only the sender shall verify suppressed bytes. If verification fails, the Payload Header shall NOT be suppressed.

11.4.9.3.7.6 Vendor-specific PHS parameters

This allows vendors to encode vendor-specific PHS parameters. The Vendor ID shall be the first TLV embedded inside Vendo-specific PHS Parameters. If the first TLV inside Vendor-specific PHS Parameters is not a Vendor ID, then the TLV shall be discarded.

Type	Length	Value
[24/25].100.12.255	*n*	—

11.4.9.4 ATM CS Encodings for Configuration and MAC-Layer Messaging

The following TLV encodings shall be used in the configuration file, in SS registration requests (when applicable), and in Dynamic Service Messages (when applicable). All ATM specific TLVs are prefixed to begin with a Type value of 99.2.

11.4.9.4.1 ATM switching encoding

This field defines the switching methodology for the service. If the field = 0, at least one VPI/VCI Classifier pair shall be defined for classifying the service. If the field = 1, exactly one VPI Classifier and zero or one VCI Classifier shall be specified for classifying the service. If the field = 2, exactly one VPI Classifier and one VCI Classifier shall be defined for classifying the service. If the field = 0, PHS is not allowed and the SDU size TLV shall equal 52. If the field = 1 and PHS is on for the service, the SDU size TLV shall equal 51; otherwise it shall be set equal to 52. If the field = 2 and PHS is on for the service, the SDU size TLV shall equal 49; otherwise it shall be set equal to 52.

Type	Length	Value
[24/25].99.0	1	0 = no switching methodology applied 1 = VP switching 2 = VC switching

11.4.9.4.2 VPI classifier

This field defines the VPI on which to classify ATM cells for the service flow.

Type	Length	Value
[24/25].99.1	2	8 or 12-bit VPI field value

11.4.9.4.3 VCI classification

This field defines the VCI on which to classify ATM cells for the service flow.

This TLV shall immediately follow the VPI TLV with which it is associated.

Type	Length	Value
[24/25].99.2	2	16-bit VCI field value

11.4.10 HMAC Tuple

This parameter contains the HMAC Key Sequence Number concatenated with an HMAC digest used for message authentication. The HMAC Key Sequence Number is stored in the four least significant bits of the first byte of the HMAC Tuple, and the most significant four bits are reserved. The HMAC-Tuple attribute format is shown in Table 142 and Table 143.

Table 142—HMAC Tuple definition

Type	Length	Value	Scope
27	21	SeeTable 143.	DSx-REQ, DSx-RSP, DSx-ACK, REG-REQ, REG-RSP, RES-CMD, DREG-CMD, TFTP-CPLT

Table 143—HMAC Tuple value field

Field	Length	Note
reserved	4 bits	
HMAC Key Sequence Number	4 bits	
HMAC digest	160 bits	HMAC with SHA-1

11.4.11 Vendor-specific information

Vendor-specific information for SSs, if present, shall be encoded in the vendor specific information field (VSIF) (type 43) using the Vendor ID field (11.4.3) to specify which tuples apply to which vendor's products. The Vendor ID shall be the first TLV embedded inside VSIF. If the first TLV inside VSIF is not a Vendor ID, then the TLV shall be discarded.

This configuration setting may appear multiple times. The same Vendor ID may appear multiple times. This configuration setting may be nested inside a Packet Classification Configuration Setting, a Service Flow Configuration Setting, or a Service Flow Response. However, there shall not be more than one Vendor ID TLV inside a single VSIF.

Type	Length	Value
43	*n*	Per vendor definition

Example:

Configuration with vendor A specific fields and vendor B specific fields:

VSIF (43) + *n* (number of bytes inside this VSIF)
8 (Vendor ID Type) + 3 (length field) + Vendor ID of Vendor A
Vendor A Specific Type #1 + length of the field + Value #1
Vendor A Specific Type #2 + length of the field + Value #2

VSIF (43) + *n* (number of bytes inside this VSIF)
8 (Vendor ID Type) + 3 (length field) + Vendor ID of Vendor B
Vendor B Specific Type + length of the field + Value

11.4.12 Confirmation code

The Confirmation Code (CC) indicates the status for the dynamic service (DSx-xxx) messages.

The CC values are specified in Table 144.

Table 144—Confirmation code

Confirmation Code	Status
0	OK/success
1	reject-other
2	reject-unrecognized-configuration-setting
3	reject-temporary / reject-resource
4	reject-permanent / reject-admin
5	reject-not-owner
6	reject-service-flow-not-found
7	reject-service-flow-exists
8	reject-required-parameter-not-present
9	reject-header-suppression
10	reject-unknown-transaction-id
11	reject-authentication-failure
12	reject-add-aborted
13	reject-exceeded-dynamic-service-limit
14	reject-not-authorized-for-the-requested-SAID
15	reject-fail-to-establish-the-requested-SA

12. System profiles

This clause defines system profiles that list sets of features and functions to be used in typical implementation cases.

12.1 Basic ATM system profile

The basic ATM system profile is intended to address the requirements of a basic ATM interfacing to an ATM backhaul on the BS side and carrying ATM cells across the air interface between the BS and SSs. It differs from a full implementation of ATM in that it assumes all data is carried in VC-switched PVCs. As such, it has the following feature sets.

For the CSs:

- PVCs are mandatory, SVCs and soft PVCs (i.e., ATM signaling support) are optional.
- VC-switched connections are mandatory, VP-switched connections are optional.

- ATM payload header suppression is mandatory as a capability, but may be turned on or off on a per connection basis.
- Only enough of the packet CS need be implemented to support the secondary management channel.

For the MAC CPS:

- Packing of multiple ATM cells into a single MAC PDU is mandatory as a capability, but may be turned on or off on a per connection basis.
- Fragmentation of SDUs on ATM traffic connections is not required, although fragmentation on the primary and secondary management channels is still required. (Note that fragmentation and packing on the same ATM connection requires that the ATM connection be treated as a variable-length packet connection.)
- ARQ is optional.
- CRC is optional.

Annex A: Bibliography

(informative)

[B1] ATM Forum Specification af-pnni-0055.000, Private Network-Network Interface Specification, Version 1.0, March 1996.

[B2] ETSI Technical Report TR 101 177 V1.1.1, Broadband Radio Access Networks (BRAN); Requirements and architectures for broadband fixed radio access networks (HIPERACCESS), May 1998.

[B3] FIPS 140-2, Security Requirements for Cryptographic Modules, May 2001.

[B4] IEEE 100™, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition.

[B5] IEEE 802.1F™-1993, IEEE Standards for Local and Metropolitan Area Networks: Common Definitions and Procedures for IEEE 802 Management Information.

[B6] IEEE 802.5™-1998, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 5: Token ring access method and physical layer specifications

[B7] IEEE 802.10™-1998, IEEE Standards for Local and Metropolitan Area Networks: Standard for Interoperable LAN/MAN Security (SILS).

[B8] IEEE 802.10c™-1998, Interoperable LAN/MAN Security (SILS)—Key Management (Clause 3).

[B9] IEEE Std 802.11™-1997, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[B10] IETF RFC 1750, “Randomness Recommendations for Security,” D. Eastlake, S. Crocker, J. Schiller, December 1994.

[B11] ISO/IEC 7498-1, Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model, 1994.

[B12] ITU-R Recommendation P.452, Prediction procedure for the evaluation of microwave interference between stations on the surface of the Earth at frequencies above about 0.7 GHz.

[B13] ITU-R Recommendation P.530-8, Propagation data and prediction methods required for the design of terrestrial line-of-sight systems.

[B14] ITU-R Recommendation F.1499, Radio transmission systems for fixed broadband wireless access (BWA) based on cable modem standards.

[B15] Milewski, A., “Periodic Sequences with Optimal Properties for Channel Estimation and Fast Start-Up Equalization,” *IBM J. Res. Develop.*, Vol. 37, No.5, Sept. 1983, pp. 426-431.

[B16] SCTE DSS 00-05 [DOCSIS SP-RFIv1.1-I05-000714], Radio Frequency Interface 1.1 Specification, July 2000.

[B17] SCTE DSS 00-09, Baseline Privacy Plus Interface Specification, December 2000.

[B18] Schneier, Bruce, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd Edition, John Wiley & Sons, 1994.