ANSI/IEEE Std 802.1B, 1995 edition

(Incorporating ANSI/IEEE Stds 802.1B-1992
and 802.1k-1993)

(Adopted by ISO/IEC and redesignated as
ISO/IEC 15802-2: 1995)

Information technology—
Telecommunications and information exchange between systems—
Local and metropolitan area networks—
Common specifications—

Part 2: LAN/MAN management

**Adopted by the ISO/IEC and redesignated as
ISO/IEC 15802-2: 1995**

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Abstract: Services and protocol elements that permit the exchange of management information between stations attached to ISO/IEC standard local and metropolitan area networks are defined. The standard includes the specification of managed objects that permit the operation of the protocol elements to be remotely managed. In addition, an architecture for station discovery and the dynamic control of event forwarding is defined. Services and protocols that support station discovery and the dynamic control of event forwarding are defined.
Keywords: event forwarding; local area networks, management; metropolitan area networks, management

---

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

ISBN 1-55937-501-9

DATE TBD *SH94259*

ANSI/IEEE Std 802.1B, 1995 Edition

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. 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.

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.

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 all 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 technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

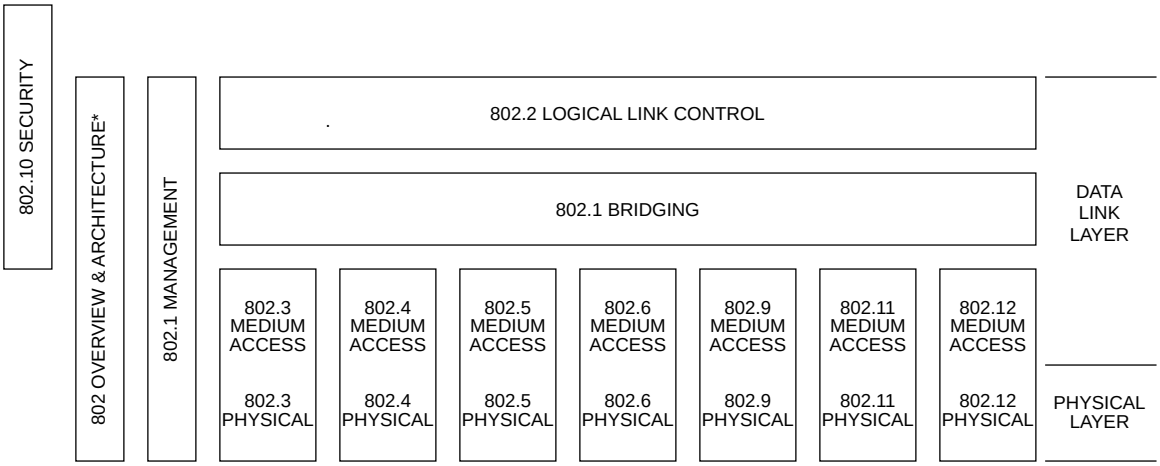
Comments on standards and requests for interpretations should be addressed to:

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

Foreword to ANSI/IEEE Std 802.1B, 1995 Edition

(This foreword is not a part of ANSI/IEEE Std 802.1B, 1995 Edition.)

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 standard numbers.)



* 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 7498 : 1984). 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[1]:	Overview and Architecture. This standard provides an overview to the family of IEEE 802 Standards. This document forms part of the 802.1 scope of work.
• ANSI/IEEE Std 802.1B [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.
• ANSI/IEEE Std 802.1D [ISO/IEC 10038]:	MAC Bridging. Specifies an architecture and protocol for the interconnection of IEEE 802 LANs below the MAC service boundary.
• ANSI/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.

[1]The 802 Architecture and Overview standard, 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.

- ANSI/IEEE Std 802.2 [ISO/IEC 8802-2]: Logical Link Control
- ANSI/IEEE Std 802.3 [ISO/IEC 8802-3]: CSMA/CD Access Method and Physical Layer Specifications
- ANSI/IEEE Std 802.4 [ISO/IEC 8802-4]: Token Bus Access Method and Physical Layer Specifications
- ANSI/IEEE Std 802.5 [ISO/IEC 8802-5]: Token Ring Access Method and Physical Layer Specifications
- ANSI/IEEE Std 802.6 [ISO/IEC 8802-6]: Distributed Queue Dual Bus Access Method and Physical Layer Specifications
- IEEE Std 802.9: Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers
- IEEE Std 802.10: Interoperable LAN/MAN Security, *Currently approved:* Secure Data Exchange (SDE)

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 following additional working groups have authorized standards projects under development:

- IEEE 802.11: Wireless LAN Medium Access Control (MAC) Sublayer and Physical Layer Specifications
- IEEE 802.12: Demand Priority Access Method/Physical Layer Specifications

Conformance test methodology

An additional standards series, identified by the number 1802, has been established to identify the conformance test methodology documents for the 802 family of standards. Thus the conformance test documents for 802.3 are numbered 1802.3, the conformance test documents for 802.5 will be 1802.5, and so on. Similarly, ISO will use 18802 to number conformance test standards for 8802 standards.

ANSI/IEEE Std 15802-2 : 1995 Edition

This document defines services and protocol elements that permit the exchange of management information between stations attached to IEEE 802 local and metropolitan area networks. The standard includes the specification of managed objects that permit the operation of the protocol elements to be remotely managed.

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

This standard contains state-of-the-art material. The area covered by this standard is undergoing evolution. Revisions are anticipated within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material. Information on the current revision state of this and other IEEE 802 standards may be obtained from

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

IEEE 802 committee working documents are available from

IEEE Document Distribution Service
AlphaGraphics #35 Attn: P. Thrush
10201 N. 35th Avenue
Phoenix, AZ 85051
USA

Participants

The following is a list of participants in the Network Management effort of the IEEE Project 802 Working Group at the time of 802.1B's approval. Voting members at the time of publication are marked with an asterisk (*). Those who were participants at the time of 802.1k's approval are marked with a dagger (†).

William P. Lidinsky, *Chair*†*
Tony Jeffree, *Chair, Network Management Task Group*†*

Fumio Akashi
Paul D. Amer
Charles Arnold
Naharaj Arunkumar
Floyd Backes*†
Ann Ballard
Richard Bantel
Robert Barrett*†
David Bartolini
Sy Bederman
Amatzia Ben-Artzi†
Anthony Berent†
Orna Berry*†
Robert Bledsoe
Kwame Boakye
Laura Bridge*†
Brian Brown†
Juan Bulnes
Fred Burg
Peter Carbone
Alan Chambers*†
Ken Chapman
Alice Chen
Michael Chernick
Jade Chien
Steve Cooper*†
Jim Corrigan
Paul Cowell*†
Mike Coy†
Andy Davis*†
Peter Dawe
Stan Degen
Frank Deignan
Desh Deshpande
Ron Dhondy
Mike Dickerson
Kurt Dobbins
Eiji Doi
Barbara J. Don Carlos
David Dyer-Bennet
Walter Eldon
Eldon D. Feist
Len Fishler*†
Kevin Flanagan
Bill Futral*†
Lionel Geretz*†
Richard Gilbert*†
Harry Gold†
Pat Gonia
Kathy de Graaf
Rich Graham
Michael A. Gravel
Andrew Green†
Sharam Hakimi*†
Jeanne Haney†
Mogens Hansen
Harold Harrington
John Hart*†
Mike Harvey†
Bob Herbst
Long Huang†
Jack R. Hung
Thomas Hytry
Jay Israel
Jan-Olof Jemnemo*†
Albert Juandy†
George Kajos†
Ram Kedlaya
Hal Keen*†
Alan Kirby
Kimberly Kirkpatrick
Steve Kleiman
Yoav Kluger†
James Kristof†
Hans Lackner*†
H. Eugene Latham
Choon Lee†
Chao-yu Liang
Bing Liao
George Lin*†
Mike Lumpkin
Andy Luque
Phil Magnuson
Joseph E. Massery†
Bruce McClure
Tom McGowan
Margaret A. Merrick
Jim Montrose
Jerry O'Keefe
Alan Oppenheimer*†
Richard Patti*†
Dave T. Perkins†
Roger Pfister
Thomas L. Phinney
Clive Philbrick
John Pickens*
David Piscitello
Daniel Pitt
Vencat Prasad*†
Ronald Presti†
Ron L. G. Prince
Maurice Qureshi†
Nigel Ramsden
Rich Rehberg
Jim Reinstedler
Trudy Reusser
Eduard Rocher
Paul Rosenblum*†
Paul Ruocchio*†
Tom Rutt*†
John Salter
Alan Sarsby
Susan Schanning
Mick Seaman*†
Gerry Segal*†
Rich Seifert*†
Steve Senum*†
Himanshu Shah*†
Howard Sherry
Wu-Shi Shung
W. Earl Smith*†
Mike Soha
Dan Stokesberry
Lennart Swartz
Kenta Takumi
Elysia Chiaw-Meng Tan
Robin Tasker*†
Angus Telfer
Dave Thompson
Geoff Thompson†
Nathan Tobol
Wendell Turner
Peter Videcrantz*†
Donald G. Vincent†
Paul Wainright
Trevor Warwick†
Scott Wasson
Bob Watson
Richard Watson*
Daniel Watts
Alan Weissberger
Deborah Wilbert
Bert Williams†
Jerry A. Wyatt†
Amnon Yacoby*†
Igor Zhovnirovsky
Carolyn Zimmer*†
Nick Zucchero

Additional participants in the development of 802.1k included the following:

Sai Boeker
Mike Dickerson
Bonnie B. Hromis
Brian J. Phillips

The following persons were on the balloting committee of 802.1B. Those who also balloted 802.1k are marked with an asterisk.

W. B. Adams*
D. Aelmore*
H. Alkhatib
K. Athul*
J. D. Brock
P. Campbell*
B. J. Casey*
A. Castaldo
K. Chon*
R. Ciciliani
M. H. Coden*
R. Crowder*
L. F. M. De Moraes
A. M. Dunn
P. Eastman*
L. G. Egan
J. E. Emrich*
P. H. Enslow*
C. Fan*
J. W. Fendrich*
H. C. Folts*
H. A. Freeman*
I. Fromm*
G. Fullerton*
P. Fung
R. Gagliano*
W. W. Garman
I. Ghansah
P. Gonia*
A. W. Hathaway
P. L. Hutton
R. J. Iliff
A. A. Jeffree*
J. R. Johnson
R. Juvonen
K. H. Kellermayr*
G. C. Kessler*
R. W. Klessig
J. Y. Lee*
F. C. Lim
R. S. Little*
J. Loo
D. C. Loughry*
N. C. Low*
W. Lu*
G. Luderer
J. F. P. Luhukay*
A. J. Luque*
K. G. McDonald
W. McDonald*
R. H. Miller*
D. S. Millman*
C. B. M. Mishra*
K. Mori*
G. Moseley
A. C. Nigam
E. S. Nolley*
D. O'Mahony*
C. Oestereicher*
Y. Oh*
A. J. Pina*
U. W. Pooch*
V. Punj*
A. Putnins*
T. L. D. Regulinski*
G. S. Robinson*
P. T. Robinson*
D. Rosich*
V. Rozentouler
D. J. Rypka
D. Sanford
R. Sankar
J. G. Sanz
B. P. Schanning*
C. Scheel
N. Schneidewind
G. D. Schumacher
J. R. Schwab*
A. S. Sethi*
D. A. Sheppard*
L. Sintonen
H. P. Solomon*
C. M. Stillebroer*
F. J. Strauss*
E. Sykas*
A. N. Tantawy
P. Thaler
G. O. Thompson*
B. A. Trent*
R. Tripi*
M. Uchida*
L. D. Umbaugh*
C. M. Weaver, Jr.
D. F. Weir*
A. J. Weissberger
R. Wenig*
E. J. Whitaker*
P. A. Willis*
J. A. Wyatt
O. Yuen*
W. Zhao

In addition to those indicated above, the following persons were on the balloting committee of 802.1k:

R. M. Amy
W. E. Ayen
M. Diaz
J. Gonzalez Sanz
C. Guarnieri
L. M. Lam
G. Lau
D. B. McIndoe
W. H. L. Moh
J. E. Montague
D. T. Perkins
J. Pickens
P. K. Piele
E. J. Reilly
R. Rosenthal
F. Ross
C. Spurgeon
J. T. Vorhies
A. D. Waren

When the IEEE Standards Board approved 802.1B on September 17, 1992, it had the following membership:

Marco W. Migliaro, *Chair* **Donald C. Loughry,** *Vice Chair*
Andrew G. Salem, *Secretary*

Dennis Bodson
Paul L. Borrill
Clyde R. Camp
Donald C. Fleckenstein
Jay Forster*
David F. Franklin
Ramiro Garcia
Thomas L. Hannan
Donald N. Heirman
Ben C. Johnson
Walter J. Karplus
Ivor N. Knight
Joseph L. Koepfinger*
Irving Kolodny
D. N. "Jim" Logothetis
Lawrence V. McCall
Don T. Michael*
L. John Rankine
Wallace S. Read
Ronald H. Reimer
Gary S. Robinson
Martin V. Schneider
Terrance R. Whittemore
Donald W. Zipse

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
James Beall
Richard B. Engelman
David E. Soffrin
Stanley I. Warshaw

Kristin M. Dittmann
IEEE Standards Project Editor

When the IEEE Standards Board approved 802.1k on June 17, 1993, it had the following membership:

Wallace S. Read, *Chair* **Donald C. Loughry,** *Vice Chair*
Andrew G. Salem, *Secretary*

Gilles A. Baril
Clyde R. Camp
Donald C. Fleckenstein
Jay Forster*
David F. Franklin
Ramiro Garcia
Donald N. Heirman
Jim Isaak
Ben C. Johnson
Walter J. Karplus
Lorraine C. Kevra
E. G. "Al" Kiener
Ivor N. Knight
Joseph L. Koepfinger*
D. N. "Jim" Logothetis
Don T. Michael*
Marco W. Migliaro
L. John Rankine
Arthur K. Reilly
Ronald H. Reimer
Gary S. Robinson
Leonard L. Tripp
Donald W. Zipse

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
James Beall
Richard B. Engelman
David E. Soffrin
Stanley I. Warshaw

Kristin M. Dittmann
IEEE Standards Project Editor

IEEE Std 802.1B-1992 was approved by the American National Standards Institute on February 23, 1993.
IEEE Std 802.1k-1993 was approved by the American National Standards Institute on January 4, 1994.

Contents

Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Common specifications—

Part 2: LAN/MAN management

1. Scope

This International Standard defines an Open Systems Interconnection (OSI) management-compatible architecture, and service and protocol elements for use in a LAN/MAN environment for the purpose of performing remote management of LAN-based or MAN-based devices. The protocol described is a connectionless-mode management protocol that makes use of Logical Link Control (LLC) Type 1 procedures as a means of conveying management information between stations in a LAN/MAN environment, thus providing for interworking of ISO/IEC standard LAN/MAN devices for management purposes. The management information is conveyed using the protocol data unit (PDU) formats defined in ISO/IEC 9596-1 : 1991.[1] To this end, this International Standard

a) Describes the services required for the transfer of management information between management processes in LAN/MAN stations.
b) Defines the protocol and PDUs for conducting management information exchanges that support the provision of those services. This protocol is defined as an (N)-layer management protocol specific to the management of layers 1 and 2 in a LAN/MAN environment.
c) Defines a convergence protocol and PDUs used for the exchange of management PDUs.
d) Describes the underlying services required for transfer of management PDUs between peer-management entities by means of the convergence protocol.
e) Defines an access control mechanism and an event report forwarding mechanism that operate in conjunction with the management protocol.
f) Defines managed object classes that relate to the operation of the management protocol and the convergence protocol.

This International Standard provides the PICS proforma for the System Load Protocol in compliance with the relevant requirements, and in accordance with the relevant guidance, given in ISO/IEC 9646-2 : 1991.

NOTES
1—This International Standard provides only a set of management tools. The management operations that are specified by this International Standard are only meaningful in conjunction with the managed object definitions contained in the appropriate layer standards.
2—This International Standard defines an (N)-layer management protocol for the management of stations attached to ISO/IEC standard LAN or MAN subnetworks, in accordance with the terminology contained in the OSI Management Framework, ISO/IEC 7498-4 : 1989 and the OSI Systems Management Overview, ISO/IEC 10040 : 1992. The service and protocol described in this International Standard are based upon the OSI Common Management Information Service and Protocol standards (ISO/IEC 9595 : 1991 and ISO/IEC 9596-1 : 1991), and are therefore designed to be used in conjunction with managed objects defined in accordance with the Guidelines for the definition of managed objects (ISO/IEC 10165-4 : 1992). As such, this International Standard is intended to be complementary to the functionality of the OSI Management standards being developed in ISO/IEC JTC1 SC21/WG4.

[1] Information on references can be found in clause 2.

3—To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given protocol. Such a statement is called a Protocol Implementation Conformance Statement (PICS). Annex A to this International Standard contains the PICS proforma for the LAN/MAN Management Protocol.

2. References

The following standards contain provisions which, through references in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

IEEE Std 802-1990, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture (ANSI).[2]

IEEE Std 802.1F-1993, IEEE Standards for Local and Metropolitan Area Networks: Common Definitions and Procedures for IEEE 802 Management Information (ANSI).

IEEE Std 802.10-1992, IEEE Standards for Local and Metropolitan Area Networks: Standard for Interoperable LAN/MAN Security (SILS), *Currently Contains* Secure Data Exchange (SDE) (Clause 2) (ANSI).

ISO 6093 : 1985, Information processing systems—Representation of numerical values in character strings for information interchange.[3]

ISO 7498 : 1984, Information processing systems—Open Systems Interconnection—Basic Reference Model.

ISO/IEC 7498-4 : 1989, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 4: Management framework.

ISO/IEC 8802-2 : 1994 [ANSI/IEEE Std 802.2, 1994 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

ISO/IEC 8824 : 1990, Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1).

ISO/IEC 8825 : 1990, Information technology—Open Systems Interconnection—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

ISO/IEC 9595 : 1991, Information technology—Open Systems Interconnection—Common management information service definition.

ISO/IEC 9596-1 : 1991, Information technology—Open Systems Interconnection—Common management information protocol—Part 1: Specification.

ISO/IEC 9596-2 : 1993, Information technology—Open Systems Interconnection—Common management information protocol—Part 2: Protocol Implementation Conformance Statement (PICS) proforma.

ISO/IEC 9646-1 : 1991, Information technology—Open Systems Interconnection—Conformance testing methodology and framework—Part 1: General concepts.

---

[2]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

[3]ISO and ISO/IEC publications are available from the ISO Central Secretariat, 1 rue de Varembé, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse. In the US, they are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

ISO/IEC 9646-2 : 1991, Information technology—Open Systems Interconnection—Conformance testing methodology and framework—Part 2: Abstract test suite specification.

ISO/IEC 10038 : 1993 [ANSI/IEEE Std 802.1D, 1993 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Media access control (MAC) bridges.

ISO/IEC 10040 : 1992, Information technology—Open Systems Interconnection—Systems management overview.

ISO/IEC 10164-1 : 1993, Information technology—Open Systems Interconnection—Systems management—Part 1: Object Management Function.

ISO/IEC 10164-1 : 1993, Information technology—Open Systems Interconnection—Systems management—Part 2: State Management Function.

ISO/IEC 10164-3 : 1993, Information technology—Open Systems Interconnection—Systems management—Part 3: Attributes for Representing Relationships.

ISO/IEC 10164-4 : 1992, Information technology—Open Systems Interconnection—Systems management—Part 4: Alarm Reporting Function.

ISO/IEC 10164-5 : 1993, Information technology—Open Systems Interconnection—Systems management—Part 5: Event Report Management Function.

ISO/IEC 10164-6 : 1993, Information technology—Open Systems Interconnection—Systems management—Part 6: Log Control Function.

ISO/IEC 10164-7 : 1992, Information technology—Open Systems Interconnection—Systems management—Part 7: Security Alarm Reporting Function.

ISO/IEC 10165-1 : 1993, Information technology—Open Systems Interconnection—Management information services—Structure of management information—Part 1: Management Information Model.

ISO/IEC 10165-2 : 1992, Information technology—Open Systems Interconnection—Management information services—Structure of management information—Part 2: Definition of management information.

ISO/IEC 10165-4 : 1992, Information technology—Open Systems Interconnection—Management information services—Structure of management information—Part 4: Guidelines for the definition of managed objects.

ISO/IEC 10742 : 1994, Information technology—Telecommunications and information exchange between systems—Elements of management information related to OSI Data Link Layer standards.

ISO/IEC 15802-4 : 1994 [ANSI/IEEE Std 802.1E, 1994 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 4: System load protocol.

ISO/IEC TR 10178 : 1992, Information technology—Telecommunications and information exchange between systems—The structure and coding of Logical Link Control addresses in Local Area Networks.

ISO/TR 8509 : 1987, Information processing systems—Open Systems Interconnection—Service conventions.

ITU-T Recommendation X.219 (1988), Remote Operations: Model, Notation and Service Definition, *Blue Book*, Vol. VIII.4.[4]

[4]ITU-T publications are available from the International Telecommunications Union, Sales Section, Place des Nations, CH-1211, Genève 20, Switzerland/Suisse. They are also available in the United States from the U.S. Department of Commerce, Technology Administration, National Technical Information Service (NTIS), Springfield, VA 22161, USA.

3. Definitions

3.1 Definitions related to local and metropolitan area networks

Many of the specialized terms used in this International Standard are defined in IEEE Std 802-1990.

3.2 Logical Link Control definitions

This International Standard makes use of the following terms defined in ISO/IEC 8802-2 : 1994:

a) LLC address
b) LSAP address

3.3 Basic Reference Model definitions

This International Standard makes use of the following terms defined in ISO 7498 : 1984:

a) open system
b) systems management

3.4 Management Framework definitions

This International Standard makes use of the following terms defined in ISO/IEC 7498-4 : 1989:

a) managed object
b) management information base

3.5 Systems Management Overview definitions

This International Standard makes use of the following terms defined in ISO/IEC 10040 : 1992:

a) agent
b) (N)-layer management protocol
c) managed object class
d) management information
e) manager
f) notification
g) notification type
h) (systems management) operation
i) systems managed object
j) systems management protocol

3.6 Structure of Management Information (SMI) Information Model definitions

This International Standard makes use of the following terms defined in ISO/IEC 10165-1 : 1993:

a) behaviour
b) containment
c) managed object boundary
d) name binding

e) package
f) subordinate object
g) superior object

3.7 Common Management Information Service (CMIS) definitions

This International Standard makes use of the following terms defined in ISO/IEC 9595 : 1991:

a) attribute
b) Common Management Information Services

3.8 Abstract Syntax Notation One (ASN.1) definitions

This International Standard makes use of the following terms defined in ISO/IEC 8824 : 1990:

a) object identifier
b) type

3.9 Guidelines for the Definition of Managed Objects (GDMO) definitions

This International Standard makes use of the following terms defined in ISO/IEC 10165-4 : 1992:

a) managed object class definition
b) template

3.10 Conformance testing definitions

This International Standard uses the following terms defined in ISO/IEC 9646-1 : 1991:

a) PICS proforma
b) protocol implementation conformance statement (PICS)
c) static conformance review

3.11 Terms defined in this International Standard

The following terms are used throughout this document in a specialized manner:

3.11.1 affiliate: A remote convergence protocol entity (CPE) whose CPE address is known to the local CPE.

3.11.2 affiliation: A state that exists if both remote and local CPEs know each other’s CPE addresses.

3.11.3 asynchronous: Protocol operation in which more than one exchange between a given pair of entities can be handled simultaneously.

3.11.4 convergence protocol: A protocol that provides the convergence service.

3.11.5 convergence service: A service that provides enhancements to an underlying service in order to provide for the specific requirements of the convergence service user.

3.11.6 CPE address: The LSAP address at which the CPE may be reached.

3.11.7 CPE instance identifier: The tuple of CPE address and CPE instance number that uniquely identifies a CPE instance within the LAN/MAN environment, within the limits of uniqueness of the CPE address and instance values used.

3.11.8 CPE instance number: A number, allocated to the CPE at instantiation time, that distinguishes a CPE instance from all other CPE instances, past and present, associated with a particular CPE address.

3.11.9 LAN/MAN Management: The management functionality specific to the management of IEEE 802 Local or Metropolitan Area subnetworks.

3.11.10 resource: That part of a LAN/MAN environment for which a managed object provides the management view. The management view of a resource may be limited to a subset of the functionality of the resource; some aspects of the resource may therefore be inaccessible for management purposes.

3.11.11 synchronous: Protocol operation in which only one exchange between a given pair of entities can be handled at any moment in time. The current exchange must complete before the next can be initiated.

3.12 Acronyms and abbreviations

The following acronyms and abbreviations are used in this International Standard:

ACF	Access Control Function
ACSE	Association Control Service Element
APDU	Application PDU
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
Conf	Confirm
CP	Convergence Protocol
CPDU	Convergence Protocol Data Unit
CPE	Convergence Protocol Entity
DLSDU	Data Link Service Data Unit
DSAP	Destination SAP
GDMO	Guidelines for the Definition of Managed Objects
ID	Identifier
Ind	Indication
LAN	Local Area Network
LCI	Local CPE Instance
LLC	Logical Link Control
LMMP	LAN/MAN Management Protocol
LMM_PDU	LAN/MAN Management Protocol Data Unit
LMMPE	LAN/MAN Management Protocol Entity
LMMS	LAN/MAN Management Service
LMMU	LAN/MAN Management User
LRG	Local Request Group
LRI	Local Request Instance
LSAP	Link SAP
MAC	Media Access Control
MAN	Metropolitan Area Network
MIB	Management Information Base
OSI	Open Systems Interconnection
PDU	Protocol Data Unit

PE	Protocol Entity
PICS	Protocol Implementation Conformance Statement
QOS	Quality of Service
RCI	Remote CPE Instance
Req	Request
RO-APDU	Remote Operations APDU
ROSE	Remote Operations Service Element
RRG	Remote Request Group
RRI	Remote Request Instance
Rsp	Response
SAP	Service Access Point
SDU	Service Data Unit
SMI	Structure of Management Information
SNAP	Subnetwork Access Protocol
SSAP	Source SAP
TR	Technical Report

4. LAN/MAN Management and Systems Management

OSI systems management functions are defined as those functions that support management applications, and that allow such applications to manipulate resources in the OSI environment. The management functional areas that give rise to these functions are described in the OSI Management Framework (ISO/IEC 7498-4 : 1989) and Systems Management Overview (ISO/IEC 10040 : 1992). The current set of functions defined by ISO/IEC are to be found in the Systems Management Function standards (ISO/IEC 10164, Parts 1–7).

The OSI Management Framework (ISO/IEC 7498-4 : 1989) defines Systems Management as an Application layer activity, related to the management of the use of OSI resources and their status across all seven layers of the OSI architecture. As a consequence, Systems Management requires the use of Application layer protocols supported by a full seven-layer protocol stack. The OSI Systems Management functions therefore require the support of CMIS, CMIP, and ROSE [ISO/IEC 9595 : 1991, ISO/IEC 9596-1 : 1991, and ITU-T Recommendation X.219 (1988)], which in turn require the support of a full connection-oriented OSI stack.

Typical systems management activities include

a) Management of the activation and deactivation of OSI resources
b) Software loading
c) Reporting status, status changes, and statistics
d) Error detection, diagnosis, and recovery

The requirements for Systems Management protocol support can, however, be in conflict with the general requirements for management of the communications environment in that

— Management of the communication capability of a station is often required when part of the communication capability is unavailable, inoperable, or approaching inoperability. This may require the management protocol to be carried directly by simple, lower-layer services. Examples of such requirements are initialization and loading of a station’s system software during “boot-strapping” of the system.
— Many of the devices that make up a communications network are constrained by memory or other resource limitations in ways that prohibit support of a full seven-layer connection-oriented OSI protocol stack for management purposes (e.g., modems, link layer bridges, repeaters, and hubs), and more economical methods of operation are a practical necessity.

The OSI Management Framework (ISO/IEC 7498-4 : 1989) allows for the provision of management functionality by means of (N)-layer management protocols in circumstances where the services of all seven layers are not available and where management is performed entirely within the context of specific layers. The IEEE 802.1 LAN/MAN Management protocol is therefore defined as an (N)-layer management protocol specific to the management of layers 1 and 2 in a LAN/MAN environment. It provides a means for resource-constrained, simple, or broken systems to be managed in a manner that is better suited to the capabilities of such systems while allowing them to interoperate in the OSI environment. This is achieved by defining the LAN/MAN Management protocol in such a way that it requires only the services of the lower two layers, specifically the simple connectionless services available at the link layer in LAN/MAN technologies, to support the exchange of management information between stations.

The use of the LAN/MAN Management protocol does not preclude the use of OSI systems management protocols; in particular, the concepts of managed objects and management information used in this International Standard are based on those described in the OSI Structure of Management Information standards (ISO/IEC 10165, Parts 1, 2, and 4), and the protocol defined in this International Standard makes use of the protocol data units defined in CMIP (ISO/IEC 9596-1 : 1991) to carry management information between systems. The management information exchanged between stations may therefore be defined in a manner that makes it available both to the LAN/MAN Management protocol and to CMIP (ISO/IEC 9596-1 : 1991); hence this information may also be used to support Systems Management. The manner in which this specification is accomplished is described in IEEE Std 802.1F-1993.

NOTES

1—It is recommended that, where practicable, management is performed through the use of OSI Systems Management, which operates over a full seven-layer protocol stack.

2—The management services and supporting protocols described in this International Standard were developed with the specific aim of providing remote management for layers 1 and 2 of the OSI reference model as they apply to a LAN/MAN environment. The use of these mechanisms in other contexts is outside the scope of this International Standard, but such use is in no way precluded.

5. Architecture

This clause contains a description of the major components of the LAN/MAN Management architecture and defines the terms used in relation to the architecture.

Within a LAN/MAN station, there are components that are concerned with the normal communication activities of the station, and components that are concerned with its management activities. The description in this clause relates only to

a) The management aspects of a LAN/MAN station and
b) The communication services and protocols that are required for management purposes.

NOTE—The description of the architecture makes use of the OSI concepts of a *service definition*, which defines, in an abstract manner, the mechanisms that permit the exchange of information between peer entities in the OSI environment; and of a *peer protocol*, which is used, in a defined way, to provide a given service. It should be borne in mind that service definitions do not describe exposed interfaces, and are therefore not the subject of conformance requirements.

5.1 Management communication

Figure 5-1 is an adaptation of figure A.2 in Annex A of the OSI Management Framework, ISO/IEC 7498-4 : 1989, showing how the concept of the (N)-layer management protocol is utilized in this International Standard in order to define an (N)-layer management protocol for use in the management of layers 1 and 2 in a LAN/MAN environment. The LAN/MAN Management Protocol defined by this International Standard corresponds to the (N)-layer management protocol shown in the figure; it in turn makes use of the services provided by the normal communication protocols available at layers 1 and 2 in a LAN/MAN environment.

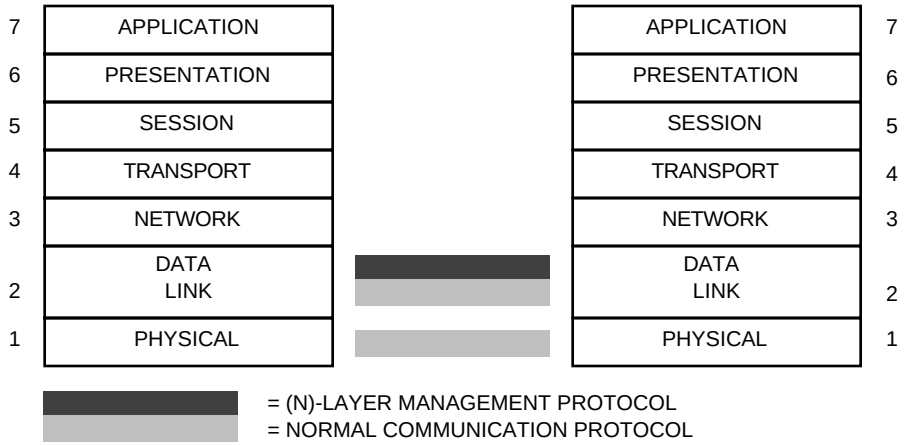


Figure 5-1—(N)-layer management protocol exchange at the Data Link layer

Figure 5-2 depicts the service and protocol elements that are involved in management communication between LAN/MAN stations by means of the LAN/MAN Management protocol. There are three major elements involved in LAN/MAN Management communication: the *LAN/MAN Management Service (LMMS)*, the *LAN/MAN Management Protocol Entity (LMMPE)*, and the *Convergence Protocol Entity (CPE)*.

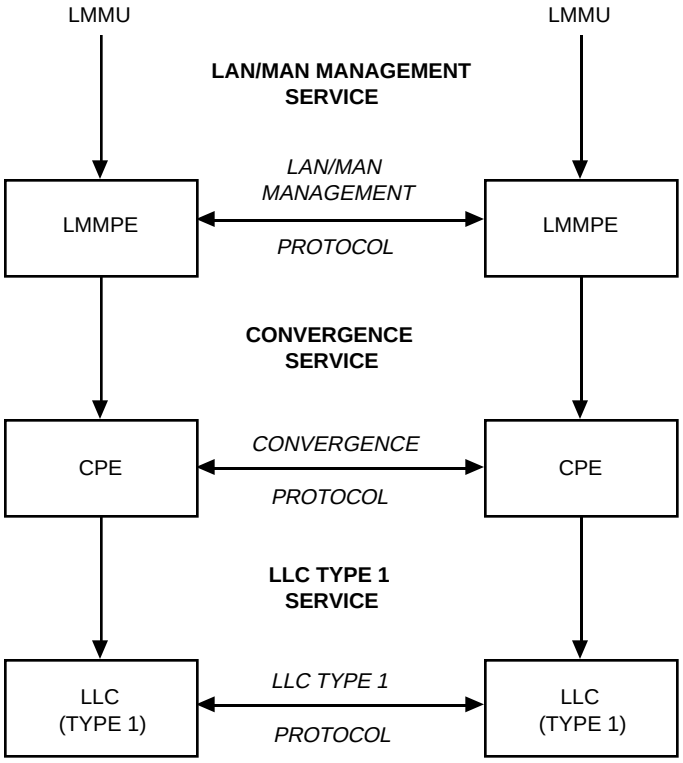


Figure 5-2—LAN/MAN Management communication architecture

The LMMS defines the service that is available to the management process known as the *LAN/MAN Management User (LMMU)* for the purposes of exchanging management information with another LMMU. The LMMS makes use of service primitives defined in the Common Management Information Service (CMIS) (ISO/IEC 9595 : 1991), in the manner defined in clause 6.

The LMMPE is the protocol entity that is responsible for providing the LMMS, by means of LAN/MAN Management protocol exchanges. The LMMS makes use of procedures and protocol data units defined in the Common Management Information Protocol (CMIP) (ISO/IEC 9596-1 : 1991), in the manner defined in clause 7.

The CPE is a protocol entity that is responsible for providing convergence functions that enable the LMMS to be provided in a LAN/MAN environment. These functions are concerned with the provision of reliability and the maintenance of logical associations between LMMUs. The convergence service and the protocol that support these functions are described in clause 7.

The CPE assumes the availability of the Data Link service as an underlying service for the exchange of LAN/MAN Management protocol data units (LMM_PDUs). Only the LLC Type 1 (unacknowledged connectionless) procedures are used to support the operation of the convergence protocol.

5.2 Management information and management operations

The component of a station that exists within a given OSI layer is known as a layer subsystem. Within a layer subsystem, resources such as protocol machines provide the normal communication functions of the layer. The definition of the functions of these entities is not within the scope of this International Standard.

For management purposes, it may be necessary to provide a means of monitoring and controlling aspects of the operation of layer subsystems and their components. This is achieved by the provision of management functionality that is concerned with providing the *management view* of the layer subsystem. This functionality provides *management information* about the layer subsystem, effects control over it in defined ways, and indicates the occurrence of certain events within it. This functionality is defined in terms of one or more *managed objects* that provide the management view of the resources of the layer subsystem.

NOTE—The OSI Management Framework (ISO 7498 : 1984) describes the concept of the *Management Information Base (MIB)*, which is a conceptual repository of management information that corresponds to the set of managed objects instantiated in a system and the information associated with them; however, the MIB is not explicitly defined as part of the 802.1 architecture, as the concept does not add value to the concepts of managed objects and management information. The concepts described here are directly compatible with the OSI concept of the MIB and the managed objects that it contains; in particular, the use of CMIP (ISO/IEC 9596-1 : 1991) PDUs as the basis for the LMMP permits managed objects defined in accordance with the GDMO (ISO/IEC 10165-4 : 1992) templates to be compatible with both the LMMP and the CMIP.

Layer subsystems are also responsible for participating in other distributed functions (e.g., dynamic routing at the network layer). In order to achieve this, specific protocols may exist for communication of management information related to those functions. This aspect of management is not within the scope of this International Standard.

The functionality of a managed object is made available via the *managed object boundary*, as expressed in the OSI Management Information Model (ISO/IEC 10165-1 : 1993). The Management Information Model defines the set of generic operation and notification types that a managed object may support; a given managed object may support a subset of these types.

NOTE—The terms *managed object* and *managed object boundary* replace the terms *layer management entity (LME)* and *layer management interface (LMI)* that were used in early drafts of this International Standard, as follows:

a) An LME corresponds to the set of managed objects that provide the management view of a layer subsystem.
b) An LMI corresponds to the functionality available at the managed object boundaries of the set of managed objects represented by an LME.

The use of the terms LME and LMI is deprecated.

The functionality of a managed object may be utilized for management purposes by a managing process. Where the managing process is local to the managed object, the mechanisms used to effect this interaction are outside the scope of this International Standard. Where the managing process is remote from the managed object with which it is required to interact, the managing process may make use of the LMMS in order to communicate with a peer-managing process in the remote station. This communication is described in terms of the roles that each managing process plays in a particular exchange of management information—either the role of a *Manager* or the role of an *Agent*. The Manager and Agent managing processes involved in such exchanges are users of the LMMS and are therefore known as *LAN/MAN Management users (LMMUs)*. The terms Manager and Agent are used for descriptive purposes only, and refer to the roles adopted in individual exchanges of management information. A managing process may be capable of adopting either role, and may therefore participate in successive management information exchanges as a Manager or as an Agent, as appropriate to the type of exchange taking place.

A Manager *requests* management operations to be performed by a remote LMMU, the Agent. An Agent *performs* the requested operations on managed objects accessible to it as though they had been requested locally, and *responds* by returning any results to the originating Manager. However, an Agent maintains complete jurisdiction over whether the requested operations are performed. The Agent may, on the basis of access control information held locally, determine that particular operations on particular managed objects may not be performed by particular Managers. The basis upon which this determination is made is described in clauses 8 and 9.

A managed object may from time to time emit notifications that contain information related to events that have occurred within the managed object. The Agent may, on the basis of event forwarding information held locally, determine whether such notification-related information should be *forwarded* to one or more Managers in the form of event reports. The basis upon which this determination is made is described in clause 8.

Figures 5-3 and 5-4 show the relationship between the Manager, the Agent, the LMMS, the LMMP, and managed objects during management information exchanges.

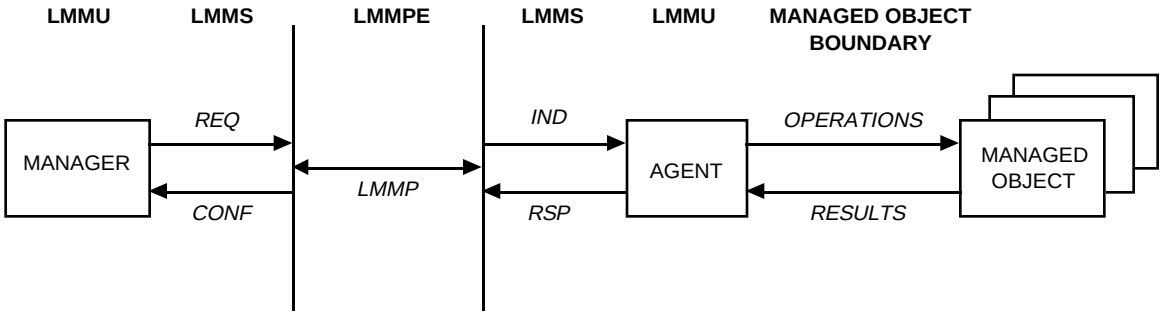


Figure 5-3—LAN/MAN Management information exchanges: operations

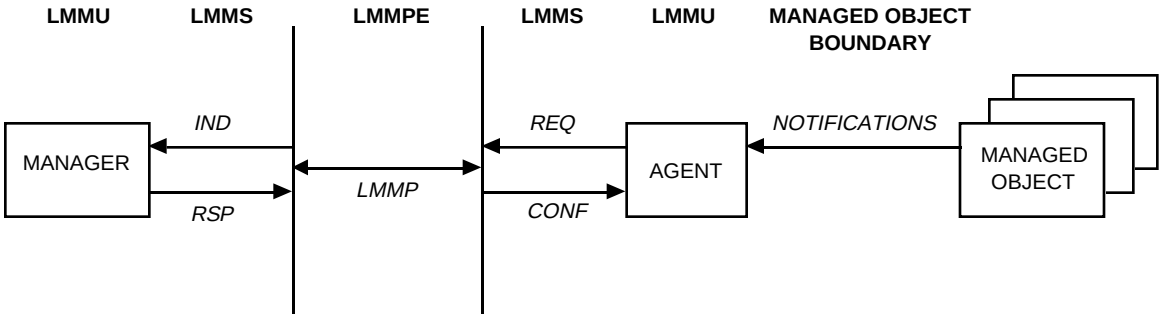


Figure 5-4—LAN/MAN Management information exchanges: notifications

In an environment where more than one Manager may exist, Managers and Agents may be grouped logically into management domains in order to determine which Manager(s) may request operations of which Agent(s), and to determine which Manager(s) an Agent should send event reports. Managed objects are defined in clause 8 for this purpose. The exact nature of management domains (whether hierarchical, exclusive, overlapping, etc.) is outside the scope of this International Standard.

The managed objects defined in clause 8 provide management functionality specific to the management of the security, event forwarding, and other management protocol-related functionality described in this International Standard, and allow these resources to be managed in a fashion analogous to the management of the resources of a layer subsystem.

5.3 Relationship with CMIS/CMIP

As indicated in 5.1, the LMMS and LMMP are realized by making use of the services, protocol data units, and procedures defined in CMIS and CMIP (ISO/IEC 9595 : 1991 and ISO/IEC 9596-1 : 1991) in the manner described in 6.1 and 7.1. This means that the functionality available from the LMMS and CMIS are the same, and that the PDUs used to express that functionality are also the same. This leads directly to the following two scenarios where CMIS/CMIP and LMMS/LMMP may be used in combination:

a) Use of CMIP and LMMP in the same Manager and/or Agent system, in order to permit management of, or management by, systems that implement only one of the two protocols.
b) “Proxy” Managers, which implement both protocols, and are capable of relaying requests between CMIP-based systems and LMMP-based systems.

In both scenarios, the compatibility between the services and protocols means that the managed object definitions are the same, regardless of the choice of protocol. Managed objects instantiated in accordance with those definitions may be accessed by either or both protocols.

NOTE—These scenarios are included for illustrative purposes only, and do not imply any conformance requirement upon the use of either CMIP or LMMP.

Figure 5-5 shows the correspondence between the protocol stacks for CMIP and the LMMP. The CPE is responsible for the control of errors, such as those arising due to lost CPDUs or misordering of CPDUs; and it relies upon the error-detection capabilities of the underlying MAC service, in particular CRC checking, for other aspects of data integrity.

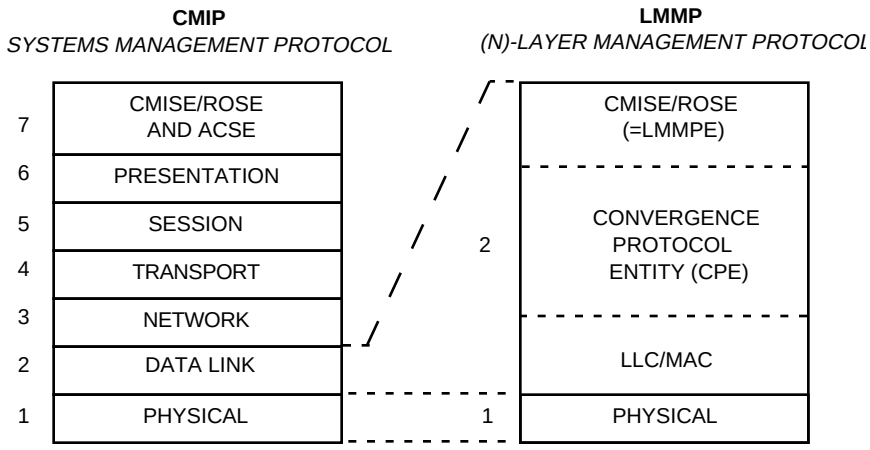


Figure 5-5—Comparison between CMIP and LMMP protocol stacks

5.4 Relationship with other management protocols

Management information defined for use with other management protocols can be accessed by use of the LMMP, provided that suitable definitions of that management information are available, expressed as managed object definitions using the notation defined in ISO/IEC 10165-4 : 1992.

6. Services

This clause defines the LAN/MAN Management service provided by the LMMPE, the Convergence service provided by the CPE, and the relationship between LMMS services and the managed object boundary.

6.1 LAN/MAN Management service

The LMMS consists of the services defined in CMIS (ISO/IEC 9595 : 1991), with the exception that the LMMS does not require the provision of association establishment or release services and places some additional constraints upon the existing service definitions in order to meet the requirements for operation in a connectionless LAN/MAN environment. As such, it relies on error reporting to indicate functionality that is not supported, as stated in the definition of the systems management application context contained in ISO/IEC 10040 : 1992. The access control parameters of the LMMS service primitives may be used to exchange access control information between Manager and Agent, for the purposes of the access control mechanisms described in 9.2. The LMMS services are summarized in table 6-1.

Table 6-1—LMMS services

Service	Type
M-ACTION	confirmed/non-confirmed
M-CANCEL-GET	confirmed
M-CREATE	confirmed
M-DELETE	confirmed
M-EVENT-REPORT	confirmed/non-confirmed
M-GET	confirmed
M-SET	confirmed/non-confirmed

6.1.1 Additional constraints on CMIS

Given that the LMMS does not require an explicit association establishment or release phase, the references to A-ABORT in clause 9 of CMIS do not apply to this International Standard.

6.2 Convergence function and convergence service

The convergence function of the CPE provides

a) The information necessary for stations participating in management information exchanges to detect station reboot or reset.
b) The information necessary for stations participating in management information exchanges to detect or determine
    1) Duplication of information

2) Out-of-sequence information
3) Temporal ordering of information
4) Information loss

c) The information necessary to identify the abstract syntax of the management information being exchanged and the LMMP protocol version.

d) Mechanisms that improve the reliability of communication for confirmed information exchanges in circumstances where the rate of packet loss in connectionless LAN/MAN data transfers is considered to be unacceptably high for management information exchanges.

Items a), b), and c) are provided by means of information that is carried in CPDUs exchanged between peer CPEs. Item d) is provided by means of a simple protocol state machine that performs timeouts and retries in conjunction with the reliable quality of service (QOS) provided by the LMMS. The details of the CPDU structure and the state machines that support these aspects of the CPE are described in clause 7.

The convergence function operates on the basis of an *affiliation* between the CPEs involved in an exchange of information; the minimum requirement for an affiliation is that the CPEs are aware of each other's *CPE instance identifiers* (see 7.3.1).

NOTES

1—The convergence protocol provides a means whereby a change in the status of an affiliation may be detected, caused by a system reboot or reset by either party. The action taken upon such events is a local matter.

The convergence function places a limit on the number of outstanding requests between affiliates.

2—CPE-Abort indications (see 6.2.3) will result if the convergence service user operates using ROSE Operation Class 2 (asynchronous), or any other analogous procedure that may result in multiple outstanding requests at the convergence service level, and the number of outstanding requests exceeds the capabilities of the affiliates concerned.

The operation of the convergence function may result in a non-negligible rate of user information loss, depending upon whether the reliable QOS is chosen, the number of retries that are configured, and the behaviour of the LAN/MAN environment in which the protocol is operated. Where the reliable QOS option is not used, reliability is directly dependent upon the reliability of the unacknowledged connectionless-mode services available in the LAN/MAN environment.

The convergence function does not provide any capability for segmentation and reassembly of user data; the maximum size of any user data is therefore constrained by the maximum frame size available in the LAN/MAN environment.

The convergence function may be viewed as a PDU delivery service for the user of the convergence service, the LMMPE. The CPE accepts PDUs from the convergence service user for delivery to a particular destination, and delivers them to the peer CPE at that destination. The receiving CPE then passes the received PDU to its service user. In addition to this delivery service, the CPE provides an abort indication service that alerts the convergence service user to service provider aborts. The service primitives associated with the convergence service are described below.

6.2.1 Conventions

This International Standard defines the services provided by the CPE following the descriptive conventions defined in ISO/TR 8509 : 1987. The definition of each CPE service includes a table that lists the parameters of its primitives. For a given primitive, the presence of each parameter is described by one of the following values:

M	The parameter is mandatory.
(=)	The value of the parameter is equal to the value of the parameter in the column to the left.
U	The use of the parameter is a service-user option.
—	The parameter is not present in the interaction described by the primitive concerned.
C	The parameter is conditional. The condition(s) are defined by the text that describes the parameter.

6.2.2 CPE-Data service

The CPE-Data service is invoked by the convergence service user issuing a CPE-Data request service primitive to the CPE, with parameters set according to the parameter definitions contained in this clause. The CPE User Information contained in the CPE-Data request primitive is delivered to the appropriate peer CPE, using the requested QOS, by means of the convergence protocol described in 7.3. The information is then passed to the peer convergence service user by the peer CPE issuing a CPE-Data request primitive, with parameters set to reflect the CPE User Information received and the actual QOS used.

The parameters associated with the CPE-Data request and indication primitives are listed in table 6-2.

Table 6-2—Parameters associated with the CPE-Data service

Parameter Name	Req	Ind
Destination Address	M	—
Source Address	—	M
QOS	M	M
Status	—	M
CPE User Information	M	M

a) **Destination Address:** Indicates to the CPE the address of the peer convergence service user to which the CPE User Information is to be delivered.

b) **Source Address:** Indicates the CPE address of the requesting convergence service user.

c) **QOS:** In request primitives, indicates to the CPE the quality of service with which the CPE User Information is to be delivered to the peer convergence service user. This parameter permits the requesting convergence service user to specify the following:

1) Priority. This value corresponds to the priority with which the CPE is requested to deliver the CPE User Information.

2) Reliability. The CPE service user may request either *basic* or *enhanced* reliability. This value determines whether the CPE will deliver the CPE User Information by making direct use of the underlying LLC service, or whether the CPE will provide enhancements that improve the reliability of the underlying service.

In indication primitives, the QOS parameter indicates the actual quality of service used for delivery of the CPE User Information.

NOTE—Requesting basic reliability will result in the use of the underlying LLC Type 1 procedures without the application of timeouts and retries. Requesting enhanced reliability will result in the use of timeouts and retries in order to recover from data loss. The Priority value is mapped directly onto the priority parameter in the DL-UNITDATA primitives.

d) **Status:** Indicates the status of the CPE from which the CPE User Information was received. The permitted values are

1) New affiliate

2) Old affiliate

3) Changed affiliate

The three affiliate types are described in 7.3.1.2.

e) **CPE User Information:** Contains the information that the requesting convergence service user requests the CPE to deliver to the peer convergence service user.

6.2.3 CPE-Abort service

If the CPE is unable to deliver CPE User Information associated with a CPE-Data service primitive to its destination, the CPE issues a CPE-Abort indication primitive to the requesting convergence service user, giving the reason for the failure and the CPE User Information that it has failed to deliver.

The parameters associated with the CPE-Abort indication primitive are listed in table 6-3.

Table 6-3—Parameters associated with the CPE-Abort service

Parameter Name	Ind
Reason	M
CPE User Information	M

a) **Reason:** Indicates the reason for the convergence service provider abort. This parameter takes one of the following values:
   1) Timeout
   2) Resource limitation
   3) Failure of underlying services

b) **CPE User Information:** Contains the information that the requesting convergence service user requests the CPE to deliver to the peer convergence service user.

6.2.4 CPE-Status service

If information held about an old affiliate changes, the CPE issues a CPE-Status indication primitive to the convergence service user.

The parameters associated with the CPE-Status indication primitive are listed in table 6-4.

Table 6-4—Parameters associated with the CPE-Status service

Parameter Name	Ind
CPE identifier	M
Status	M

a) **CPE identifier:** The instance identifier of the CPE whose status information is being reported.

b) **Status:** Indicates the status of the CPE from which the new status information was received. The permitted values are changed affiliate. The meaning of changed affiliate is defined in 7.3.1.2.

6.3 Relationship between LMMS services and the managed object boundary

6.3.1 Managed objects and the managed object boundary

The OSI Management Information Model (ISO/IEC 10165-1 : 1993) defines the architectural basis for the definition of the managed objects and the set of operations and notifications that a managed object may support. The managed object boundary is the boundary at which the management operations and notifications supported by a managed object are made visible. Each managed object supports a defined set of operations and notifications, as identified in the managed object class definition for the managed object class of which it is an instance. The operations are of two types: those that apply to attribute manipulation, and those that apply to a managed object as a whole. Notifications may be emitted as a result of changes in attribute states

or as a result of some other event occurring within the managed object. These operations and notifications are summarized in table 6-5.

Table 6-5—Management operations

Operation/Notification	Type
Get attribute value	Attribute related
Replace attribute value	" "
Set-to-default value	" "
Add member	" "
Remove member	" "
Create	Object related
Delete	" "
Action	" "
Notification	Attribute/Object related

ISO/IEC 10165-4 : 1992 defines the notation that shall be used for defining managed object classes, their attributes, behaviour, operations, and notifications. Additional information on the requirements for the definition of managed objects can be found in IEEE Std 802.1F-1993.

6.3.2 LMMS/Managed object boundary correspondence

The services of the LMMS are used to convey requests to perform management operations on managed objects and reports of notifications emitted by managed objects between peer-managing processes. The correspondence between the LMMS services and the operations and notifications available at the managed object boundary is shown in table 6-6.

Table 6-6—LMMS/Managed object boundary correspondence

Operation/Notification	LMMS service
Get attribute value	M-GET, M-CANCEL-GET
Replace attribute value	M-SET
Set-to-default value	M-SET
Add member	M-SET
Remove member	M-SET
Create	M-CREATE
Delete	M-DELETE
Action	M-ACTION
Notification	M-EVENT-REPORT

7. Protocol

This clause defines the procedures and PDU structures associated with the LAN/MAN Management Protocol (LMMP) and the Convergence Protocol and discusses the use of underlying services by the CPE.

7.1 LMMP definition

The LMMP consists of the elements of procedure and abstract syntax defined in CMIP (ISO/IEC 9596-1 : 1991), with additional restrictions defined in order to permit the use of the protocol over the unacknowledged connectionless-mode services available in LAN/MAN environments. These restrictions are described in the following subclauses.

7.1.1 Association establishment and release

The LMMP does not require an explicit association establishment phase before LMM_PDUs are exchanged; the services of ACSE are therefore not required. The provisions of CMIP (ISO/IEC 9596-1 : 1991) clauses 5.2.1, 6.1, 6.9, 6.10, and annex A therefore do not apply. The Convergence Protocol defined in 7.3 provides mechanisms that support the association requirements of the LMMS.

7.1.2 Use of operation classes

The confirmed operations of the LMMP may use ROSE Operation Class 2 (asynchronous) or ROSE Operation Class 1 (synchronous). Restricting the LMMP to the use of Operation Class 1 implies that the Cancel Get operation shall not be used.

7.2 Use of underlying services by the LMMP

The LMMP does not require the use of Presentation P-DATA service. The provisions of CMIP (ISO/IEC 9596-1 : 1991) clause 5.2.2 therefore do not apply. As there is no presentation entity, each LMMPE is responsible for the encoding and decoding of LMM_PDUs, using the ASN.1 Basic Encoding Rules (BER) as defined in ISO/IEC 8825 : 1990. The exchange of LMM_PDUs is achieved by means of the Convergence Service defined in 6.2, which is provided by the Convergence Protocol specified in 7.3. The following procedures specify the use of the convergence service by the LMMPE.

7.2.1 Sending LMM_PDUs

All LMM_PDUs are sent by a requesting LMMPE to a peer LMMPE by issuing a CPE-Data request primitive to the CPE. The Destination Address parameter is set to the CPE address of the peer LMMPE; the QOS parameter is set to the quality of service that the LMMPE requires for the exchange; and the CPE User Information parameter is set to contain the LMM_PDU to be delivered.

When the LMMPE uses the convergence service to issue an LMM-PDU that is being sent in response to a previous incoming LMM-PDU, the Destination Address parameter value is set to the value of the corresponding Source Address parameter associated with the incoming LMM-PDU, and the QOS parameter is set to a value no lower than the QOS with which the incoming LMM-PDU was received. In all other cases, the means by which the LMMPE determines the Destination Address and QOS values is a local matter.

7.2.2 Receiving LMM_PDUs

The LMMPE receives CPE-Data indication primitives from its local CPE whenever the CPE receives CPE User Information destined for the LMMPE. The LMM_PDU contained in the CPE User Information parameter is passed to the LMM protocol machine for appropriate processing. The Source Address and QOS

parameters indicate the CPE address of the peer LMMPE that issued the corresponding Request primitive and the QOS with which the LMM_PDU was delivered.

7.2.3 CPE aborts

Receipt of a CPE-Abort indication primitive by the LMMPE occurs when a previously issued CPE-Data request has failed. The action taken by the LMMPE in these circumstances is a local matter.

7.2.4 CPE status

Receipt of a CPE-Status indication primitive by the LMMPE occurs when a new or changed affiliate is detected. The action taken by the LMMPE in these circumstances is a local matter.

7.2.5 CPE user information

A CPE User Information parameter consists of the following items, as defined in CMIP (ISO/IEC 9596-1 : 1991):

— The ASN.1 OBJECT IDENTIFIER value that identifies the CMIP abstract syntax, namely:

{joint-iso-ccitt ms(9) cmip(1) cmip-pci(1) abstractSyntax(4)}

— The CMIP datatype ProtocolVersion, carrying a value indicating which version of the protocol is being exchanged. Conformance with this International Standard requires the support of version 2, namely:

ProtocolVersion::= BITSTRING {version2 (1)}

— Any valid CMIP RO-APDU, as defined according to the CMIP Abstract Syntax named above.

7.3 Convergence protocol definition

The exchange of LMM_PDUs is achieved by means of the *Convergence Protocol (CP*). As indicated in 7.4, the CPE makes use of the unacknowledged connectionless-mode services available in the LAN/MAN environment, provided by the LLC Type 1 procedures.

The protocol defines the following:

a) Procedures for exchanging CPDUs, including a simple retry mechanism to permit the provision of the reliable QOS option identified in the convergence service definition. User data provided by the convergence service user is carried, unmodified, along with additional header information, in a CPDU.
b) Data fields in the CPDU to carry the abstract syntax identification, protocol version, and protocol data unit elements of the CPE User Information parameter of the convergence service.
c) Instance and sequence numbering information to support the requirements of the convergence service.

7.3.1 Definitions

This clause defines some of the basic terminology and concepts that underpin the operation of the protocol.

7.3.1.1 CPE instance identification

Each CPE maintains a unique *CPE instance number* that is determined at CPE instantiation time (e.g., at system boot time or at any other instant in time at which the CPE is initialized) and distinguishes a particular

CPE instance from all others associated with a particular *CPE address*. The mechanism whereby the value of this instance number is determined is a local matter; however, its value shall be an integer not less than one and not exceeding 4 294 967 295.

NOTE—The intent is that the CPE instance number distinguishes the CPE instance from all other CPE instances past and present. As the CPE instance number is chosen from a finite number space, there is a finite possibility of reuse of a CPE instance number within the lifetime of instance information held in the state tables of other systems. The choice of algorithm for the selection of CPE instance numbers should therefore be made with a view to minimizing the possibility of such reuse, for example, by serial allocation of integer values or by random selection from the set of unallocated values. The lifetime of instance information held in other systems is limited by the maximum lifetime of an affiliate record, as defined in 7.3.2.2. In cases where the algorithm chosen for the allocation of CPE instance numbers results in a CPE instance number being reallocated within a time period shorter than this maximum lifetime, communication with another CPE will not be possible if the request group instance number chosen is such that CPDUs received by the remote CPE appear to be out of sequence (as defined in 7.3.4.2). This situation will persist until such a time as the state information held by the remote CPE is timed out or the CPE instance number is changed.

The CPE address is the LSAP address at which the CPE may be reached.

The tuple of CPE instance number and CPE address provides the *CPE instance identifier*, which is unique within the LAN/MAN environment, within the limits of uniqueness of the CPE address and instance values used.

The terms *remote CPE instance number (RCI)* and *local CPE instance number (LCI)* refer to CPE instance numbers for CPEs in a remote station and the local station, respectively.

7.3.1.2 Affiliation

An *affiliation* between CPEs involves the CPEs establishing sufficient information in order to successfully exchange CPDUs. The minimum requirement for affiliation is that the CPEs concerned are aware of each other's CPE instance identifiers.

A remote CPE instance that is known to a particular CPE is known as an affiliate. There are three types of affiliates:

a) *new affiliate:* A remote CPE instance identifier for which the CPE address was previously unknown to the local CPE.
b) *old affiliate:* A remote CPE instance identifier already known to the local CPE. The CPE still retains state information related to previous communications with this affiliate.
c) *changed affiliate:* A remote CPE instance identifier for which the CPE address was previously known to the local CPE, but where the remote CPE instance number is different from the one contained in the local state information. This occurs where a new CPE has been instantiated at a remote CPE address since the last communication with that CPE address.

7.3.1.3 Requests

A *request* occurs when the CPE transmits a CPDU to an affiliate, and the CPDU concerned contains CPE User Information. Requests may be confirmed, in which case they are said to be *outstanding* until such a time as a confirmation is received (or the request has timed out), or unconfirmed.

Each request is a member of a particular *request group*, corresponding to a set of requests, some or all of which may be outstanding at a particular moment in time. The operation of a particular CPE may place an upper limit on the size of a request group; this upper limit shall be not less than one and shall not exceed 256. Two identifiers are associated with requests and request groups:

a) *request group instance:* Uniquely identifies a request group comprising requests destined for a particular affiliate. A request group instance value shall be allocated to each new request group when it

is created, and shall not be reallocated during the lifetime of that request group. The value allocated by the CPE to a new request group shall be determined as follows:

— Values allocated shall be greater than zero and less than 4 294 967 296;

— The initial value allocated by the CPE (i.e., the first request group instance allocated following CPE instantiation) shall be one;

— The second and subsequent values shall be determined by incrementing the most recently allocated value by one, and if the resultant value is equal to 4 294 967 296, subtracting 4 294 967 295.

Only one request group instance is permitted per affiliate at any given instant.

A *remote request group instance (RRG)* identifies a request group instance allocated by a remote CPE instance to the local CPE instance. A *local request group instance (LRG)* identifies a request group instance allocated by the local CPE instance to a remote CPE instance.

NOTES

1—The value zero for request group instance is reserved to mean “unknown” in the case of an RRG, or “no group allocated” in the case of an LRG.

2—There is no explicit or implied relationship between requests that are in the same request group, other than that they have the same request group number.

b) *request instance:* Uniquely identifies a request within a request group. The value zero is allocated to the first request in the group; as each subsequent request is added to the group, it is allocated the smallest integer value not already allocated to another member of the group. Once the maximum value for request instance has been reached, no further requests may be issued to that affiliate until a new request group instance has been created.

A *remote request instance (RRI)* identifies a request instance allocated by a remote CPE instance to an RRG. A *local request instance (LRI)* identifies a request instance allocated by the local CPE instance to an LRG.

7.3.2 State variables

This subclause describes the state information that is maintained by the CPE in connection with the status of the CPE, its affiliates, and its outstanding requests. The information content only is described; the manner in which that information is represented or stored in a particular implementation is a local matter.

7.3.2.1 Global variables

The CPE maintains global state information as described in table 7-1. This information persists for the lifetime of the CPE.

Table 7-1—Global state variables

Variable Name	Description
Local CPE Instance (LCI)	CPE instance number of the local CPE.
Next request group	Next request group instance number available for allocation.
Max group size	Maximum number of requests in a request group, in range of 1–256.

7.3.2.2 Affiliate variables

The CPE maintains state information on a per-affiliation basis, as described in table 7-2. Each instance of this information is known as an *affiliate record*.

Table 7-2—Affiliate state variables

Variable Name	Description
Remote CPE address	CPE address of the affiliate. The MAC address component may be a group MAC address or an individual MAC address.
Local CPE address	CPE address of the local CPE. This is the CPE address known to the remote CPE; normally the MAC address component is the individual address of the local CPE, but the component may be a group MAC address if an incoming request is received using group addressing.
Remote CPE instance (RCI)	Most recent CPE instance number received from remote CPE address. Zero if not known.
Remote request group instance (RRG)	Most recent request group instance received from remote CPE address. Zero if not known.
Local request group instance (LRG)	Request group instance currently allocated to this affiliate. Zero if none allocated.
Remote request instance (RRI)	Most recent request instance received from remote CPE address for current RRG. Zero if RRG is zero.
Local request instance (LRI)	Most recently allocated request instance for the current LRG. Zero if LRG is zero.
Retry limit	Maximum number of retries permitted, per request instance, for requests to this affiliate.
Timeout	Timeout period between retries for requests to this affiliate.
Request group full	Boolean value; TRUE if local request group is full, but still contains outstanding requests.

The minimum lifetime of an affiliate record is recommended to be long enough to ensure that the record persists, beyond the last valid communication with the affiliate concerned, for a minimum time period equal to twice the product of the retry limit and timeout variables. For the purposes of determining affiliate record lifetimes, a valid communication has occurred if a CPDU has been received from the affiliate that is not out of sequence, as defined in 7.3.4.2.

It is recommended that a value of 15 seconds is used for the Timeout variable and a value of 6 is used for the Retry limit variable. With these values, the recommended minimum lifetime of an affiliate record is 3 minutes.

The maximum lifetime of an affiliate record shall be 1 hour.

NOTES

1—The choice of 15 seconds for the Timeout variable is based upon the accumulated forwarding delay that may be experienced in a maximum diameter bridged LAN configured using the parameters defined in ISO/IEC 10038 : 1993.

The actual value used for the Timeout variable in a bridged LAN should not be less than twice the maximum accumulated forwarding delay that can be experienced between any pair of stations attached to the LAN; the use of smaller values may result in a higher than necessary retransmission rate. This calculation assumes that the bridged LAN is not undergoing spanning-tree reconfiguration.

2—The maximum affiliate record lifetime determines the maximum time during which communication may be lost in the event that a procedural error or data corruption causes sequence numbers to become misordered between affiliates in a given affiliation. In circumstances where such loss of communication may be critical, a smaller value should be used for the maximum lifetime.

7.3.2.3 Request variables

The CPE maintains state information, on a per-affiliate record basis, for each outstanding request, as described in table 7-3. Each instance of this information is known as a *request record*. A request record persists from the time at which the request is issued until the last permitted retry has timed out.

Table 7-3—Request state variables

Variable Name	Description
Retry count	Count of remaining permitted retries. Zero if no more permitted.
Timeout	Remaining timeout period before next retry.
CPDU	CPDU that was issued at most recent attempt for this request.

7.3.3 Convergence protocol overview

This overview is intended to introduce the reader to the mechanisms involved in the protocol; the definitive description of the protocol is to be found in the description of the procedures and state machines.

The protocol is based on a single CPDU structure, which is used both to issue requests and to issue acknowledgments. The sequence of events is as follows:

a) A request is issued. The request contains header information reflecting the local CPE instance, the LRG and LRI that apply to the request, and the RCI, RRG, and RRI information that reflect the local CPE's knowledge of the state of the affiliate to which the request is directed. The userInfo field contains the CPE User Information to be transferred.
b) The receiving CPE receives the CPDU, passes the CPE User Information to the service user, and issues an acknowledgment to the requesting CPE if the Reliable QOS flag was asserted in the incoming request. The incoming CPDU is used to update local knowledge of the state of the requesting CPE. If it so happens that the receiving CPE has a request ready to send back, the relevant CPE User Information may be "piggybacked" onto the acknowledgment if so desired; alternatively, it may be issued as a distinct request.
c) In the event that either request or response is lost, the requesting CPE may timeout and retry the request up to a locally determined retry limit. The header information in the CPDU headers permits duplicate suppression and detection of misordering under these circumstances.

A consequence of the possibility in this protocol of constructing and sending CPDUs that carry both CPE User Information relating to new outgoing requests and confirmation information confirming previous incoming requests is that every CPDU received from an old affiliate or a changed affiliate may contain confirmation information. If an incoming CPDU contains confirmation information relating to an outstanding

request or requests, the corresponding request record or records are destroyed. This mechanism is described in detail in 7.3.4 and 7.3.5.

The protocol permits the use of group MAC addressing; however, where group MAC addressing is used, the Reliable QOS option is not available.

7.3.4 Procedures

This subclause defines procedures that are common to more than one element of the CPE state machines.

7.3.4.1 Transmission of CPDUs

CPDUs are constructed in accordance with their structural and ASN.1 definition, as identified in 7.3.6. There are three cases where CPDUs are transmitted:

a) When a CPDU is issued to carry a request that is being issued for the first time (these CPDUs may also be used to acknowledge an outstanding request received from the same affiliate).
b) When a CPDU is issued in order to repeat a request after a timeout has occurred (these CPDUs may also be used to acknowledge an outstanding request received from the same affiliate).
c) When a CPDU is issued only to acknowledge a request received from an affiliate.

The values of the fields in the CPDU are set as follows:

— The RCI, LCI, RRG, and RRI fields are set, in all cases, using the corresponding values held in the global variables and the affiliate record for the affiliate concerned.
— In case a), the User Info field is set using the CPE User Information parameter passed to the CPE in the CPE-Data request service primitive, in accordance with the requirements detailed in 7.2.5.
— In case b), the User Info field is set using the corresponding information held in the request record.
— In case c), the User Info field is absent.
— In case a), the LRG and LRI fields are set using the corresponding values held in the affiliate record for the affiliate concerned. The Reliable QOS flag is set to TRUE if the request requires acknowledgment. All other flags are set to FALSE.
— In case b), the LRG, LRI and flags fields are set using the corresponding information held in the request record.
— In case c), the LRG and LRI fields are set to zero, and all flags are set to FALSE.

The CPDU is then transmitted, using the LLC Type 1 service as described in 7.4.1, using the information held in the Remote CPE address field of the affiliate record to construct the destination_address parameter of the request primitive. The quality of service used for transmission (priority) is determined from the QOS parameter of the CPE-Data request primitive, or from the priority with which the corresponding request was received.

7.3.4.2 Reception of CPDUs

CPDUs received using the LLC Type 1 service as described in 7.4.1 and constructed in accordance with the ASN.1 definitions identified in 7.3.6 are processed by the CPE. The processing of any other PDUs received on the individual LLC address reserved for LAN/MAN Management is outside the scope of this International Standard.

Where the incoming CPDU is an acknowledgment of an outstanding request, i.e., it is from an old affiliate for which there is a request record for a request whose LRG and LRI match the RRG and RRI of the incoming CPDU, the request record for that request is discarded. If this is the last outstanding request for a request group that is marked as full, the LRG and LRI variables in the affiliate record are set to zero and the request group full flag is cleared.

Where the incoming CPDU contains a userInfo field, the value of the LRG field held in the incoming CPDU is compared with the value of RRG held in the affiliate record for the affiliate concerned (the RRG value will be zero for new affiliates or if this is the first request from an old affiliate) in order to detect out-of-sequence requests. The following algorithm is applied:

a) The RRG value is subtracted from the next expected LRG value.

b) If the resultant value is less than zero, the decimal value 4 294 967 296 is added to it.

c) If the resultant value exceeds the decimal value 2 147 483 648, the received request is defined to be out of sequence, and shall be ignored.

NOTE—In 32-bit integer arithmetic, this algorithm is equivalent to adding the two's complement of RRG to LRG; the request is out of sequence if the most significant bit of the result is set.

The source CPE address from which the CPDU was received, the User Info field of the CPDU, and the QOS with which it was received define the values of the Source Address, CPE User Information, and QOS parameters of any resultant CPE-Data indication primitive.

7.3.5 CPE state table

The state table described in table 7-4 defines the procedures involved in the transfer of all CPDUs between CPEs, and in the maintenance of the CPE state information.

Table 7-4—CPE Idle state table

STATE = CPE Idle		
Event	**Action**	**NextState**
CPE-Data request received: —Affiliate address = —individual address; —enhanced reliability QOS required; —New affiliate.	Create Affiliate record with: —RCI, RRG, RRI, LRI = 0; —Remote CPE address = affiliate address; —Local CPE address = individual address of local CPE; —LRG = Next request group. Increment Next request group. Send CPDU with User Info field set to the CPE User Information in the service primitive, and with ReliableQOS flag set to TRUE. Create request record.	CPE Idle
CPE-Data request received: —Affiliate address = group address OR enhanced reliability QOS not required; —New affiliate.	Create affiliate record with: —RCI, RRG, RRI, LRI = 0; —Remote CPE address = affiliate address; —Local CPE address = individual address of local CPE; —LRG = Next request group. Increment Next request group. Send CPDU with User Info field set to the CPE User Information in the service primitive, and with ReliableQOS flag set to FALSE. Request group full:= TRUE.	CPE Idle
CPE-Data request received: —Affiliate address = individual or group address; —Old affiliate; —Request group full = TRUE.	Reprocess CPE-Data request when all outstanding requests for this affiliate have terminated (i.e., when Request group full = FALSE).	CPE Idle

Table 7-4—CPE Idle state table (*Continued*)

STATE = CPE Idle		
Event	**Action**	**NextState**
CPE-Data request received: —Affiliate address = individual address; —enhanced reliability QOS required; —Old affiliate.	If LRG = 0; —LRG: = Next request group; —LRI: = 0; —Increment Next request group. ELSE: —Increment LRI. If LRI = Max group size -1: —Request group full:= TRUE. Send CPDU with User Info field set to the CPE User Information in the service primitive, and with ReliableQOS flag set to TRUE. Create request record.	CPE Idle
CPE-Data request received: —Affiliate address = group address OR enhanced reliability QOS not required; —Old affiliate.	If LRG = 0: —LRG:= Next request group; —LRI:= 0; —Increment Next request group. ELSE: —Increment LRI. Request group full:= TRUE. Send CPDU with User Info field set to the CPE User Information in the service primitive, and with ReliableQOS flag set to FALSE.	CPE Idle
Affiliate record exists with Request group full = TRUE; No Request records for this affiliate record.	LRG, LRI:= 0; Request group full:= FALSE.	CPE Idle
Request record exists with expired Timeout and non-zero Retry count:	Decrement Retry count; Set Timeout in request record to value held in affiliate record; Retransmit PDU.	CPE Idle
Request record exists with expired Timeout and zero Retry count:	Request Group Full:= TRUE; For each Request record outstanding for this affiliate record: —Destroy request record; —Issue CPE-Abort indication to convergence service user, with Status = Timeout.	CPE Idle
Receive CPDU: —New affiliate; —Received LRI = 0; —User Info field present.	Create affiliate record with: —Remote CPE address, Local CPE address set according to received source and destination addresses; —RCI, RRG, RRI set to received values of LCI, LRG, LRI; —LRG, LRI = 0; —Request group full = FALSE. Generate CPE-Data indication, with Status = New affiliate. If ReliableQOS flag = TRUE, send confirmation CPDU to affiliate.	CPE Idle
Receive CPDU: —Old affiliate; —No User Info field present.	If received RRG corresponds to LRG in affiliate record: —Destroy all request records for this affiliate record with LRI <= received RRI. If no remaining request records for this affiliate record: —Request group full:= TRUE.	CPE Idle

Table 7-4—CPE Idle state table (*Continued*)

STATE = CPE Idle		
Event	**Action**	**NextState**
Receive CPDU: —Old affiliate; —User Info field present.	If received RRG = LRG in affiliate record: —Destroy all request records for this Affiliate record with LRI <= received RRI. If (received LRG = RRG in Affiliate record AND received LRI is next in sequence) OR (received LRG is in-sequence and received LRI = 0): —RRG, RRI:= received LRG, LRI; —Generate CPE-Data indication, with Status = Old affiliate. If ReliableQOS flag = TRUE: —Send confirmation CPDU to affiliate. If no remaining request records for this affiliate record: —Request group full:= TRUE.	CPE Idle
Receive CPDU: —Changed affiliate; —No User Info field present.	RCI:= received LCI; RRG, RRI:= 0; If received RRG corresponds to LRG in affiliate record: —Destroy all request records for this affiliate record with LRI <= received RRI. Generate CPE-Status indication, with Status = Changed affiliate. If no remaining request records for this affiliate record: —Request group full:= TRUE.	CPE Idle
Receive CPDU: —Changed affiliate; —User Info field present.	RCI:= received LCI; RRG, RRI:= 0; If received RRG = LRG in affiliate record: —Destroy all request records for this affiliate record with LRI <= received RRI. If received LRI = 0: —RRG, RRI:= received LRG, LRI; —Generate CPE-Data indication, with Status = Changed affiliate. —If ReliableQOS flag = TRUE, send confirmation CPDU to affiliate. If no remaining request records for this affiliate record: —Request group full:= TRUE.	CPE Idle
Other:	Ignore	CPE Idle

A single-state machine is defined as the CPE Idle state machine, of which a single instance exists for the lifetime of the CPE instance. This state machine processes any incoming CPDUs and processes requests from the convergence service user. The state table defines a single state: the CPE Idle state. All events occur in this state, and, following processing of the event, the CPE returns to this state. The information held in the various state variables and records is available to, and modified by, the state machine, in accordance with the procedures described.

7.3.6 CPDUs

This subclause specifies the structure and encoding of the CPDUs exchanged between convergence protocol entities.

7.3.6.1 Transmission and representation of octets

All CPDUs shall contain an integral number of octets. The octets in a CPDU are numbered starting from 1 and increasing in the order that they are put into a Data Link Service Data Unit (DLSDU). The bits in an octet are numbered from 1 to 8, where bit 1 is the low-order bit, and the remaining bits increase in significance with each increase in bit number.

When consecutive octets are used to represent a binary number, the lower octet number has the most significant value.

All CPEs shall respect these bit- and octet-ordering conventions, thus allowing communication to take place.

7.3.6.2 CPDU structure and components

Table 7-5 describes the overall CPDU structure.

Table 7-5—CPDU structure

Component	Field Description	Octet
Header	Encoding of Identifier and Length octets of an ASN.1 IMPLICIT SEQUENCE, Tag value 8, using the indefinite length form. Hexadecimal value = A880	1, 2
	Encoding of Identifier and Length octets of an ASN.1 OCTET-STRING, primitive encoding, definite length form, length = 20 octets. Hexadecimal value = 0414.	3, 4
	Remote CPE instance number (RCI)	5–8
	Local CPE instance number (LCI)	9–12
	Remote Request Group Instance (RRG)	13–16
	Local Request Group Instance (LRG)	17–20
	Remote Request Instance (RRI)	21
	Local Request Instance (LRI)	22
	Flags	23, 24
User Info	Encoding of the ASN.1 type UserData, as defined in 7.3.6.3, using ASN.1 Basic Encoding Rules.	25–X
CPDU Terminator	End-of-contents octets, as defined by the ASN.1 Basic Encoding Rules. Hexadecimal value = 0000.	X+1, X+2

CPDUs consist of the following three components:

a) A fixed-format header, containing state and control information pertinent to the operation of the convergence protocol.
b) A variable-format User Info field, containing CPE User Information pertinent to the protocol being exchanged between peer convergence service users.

NOTE—For the purposes of this International Standard, the CPE User Information consists of CMIP RO_APDUs.

c) A PDU terminator, corresponding to the octet sequence used to terminate the indefinite length encoding of ASN.1 Basic Encoding Rules.

The definition and encoding of these components has been chosen in a manner that makes the CPDU structure compatible with the ASN.1 CPDU definition contained in 7.3.6.3 when encoded using the ASN.1 basic encoding rules (ISO/IEC 8825 : 1990). It is a conformance requirement that CPDUs shall be encoded in the manner specified in this clause.

NOTE—The restriction on the use of the Basic Encoding Rules (BER) for the encoding of the CPDU header, namely, the use of the indefinite length form for the enclosing SEQUENCE and the use of primitive encoding for the OCTET-STRING, ensures that the header size and the position of the fields within it are fixed. No restrictions apply to the use of BER for the encoding of the User Info field of the CPDU.

CPE Instance Numbers and Request Group Instance Numbers in the fixed format header may take any decimal value in the range of 0 to 4 294 967 295, in accordance with the requirements of the CPE state machines.

Request instance numbers may take any decimal value in the range of 0 to 255, in accordance with the requirements of the CPE state machines.

The Flags field consists of sixteen flags, numbered from 1 to 16, where flag 1 is encoded in the least significant bit of the field (bit 1 of octet 24), and flag 16 is encoded in the most significant bit of the field (bit 8 of octet 23). Each flag may take the Boolean value TRUE or FALSE, represented as 1 for TRUE and 0 for FALSE, in accordance with the requirements of the CPE state machines. All flags marked as RESERVED are encoded with the value FALSE. Table 7-6 shows the flag allocations.

Table 7-6—Flag allocations

Flag name and description	Flag number
Reliable QOS: A value of TRUE indicates that this request requires the use of the Reliable QOS option; i.e., requires an acknowledgment.	1
RESERVED	2
RESERVED	3
RESERVED	4
RESERVED	5
RESERVED	6
RESERVED	7
RESERVED	8
RESERVED	9
RESERVED	10
RESERVED	11
RESERVED	12
RESERVED	13
RESERVED	14
RESERVED	15
RESERVED	16

For interoperability across the set of ISO/IEC standard LAN/MAN environments (MACs), CPE implementations shall be capable of receiving CPDUs of up to and including 1500 octets in length and shall not send CPDUs of greater than 1500 octets in length.

NOTE—It may be possible in certain LAN/MAN environments to operate with CPDUs of greater than 1500 octets in length. Mechanisms for establishing whether the environment will support such sizes are outside the scope of this International Standard.

7.3.6.3 ASN.1 definition of CPDU structure

```
IEEE802-1Protocol {iso(1) member-body(2) us(840) ieee802-1B(10007) asn1Module(2) conver-
genceprotocol(0) version1(0)}
DEFINITIONS ::= BEGIN

IMPORTS

LoadPDU
FROM ieee802dot1LoadProtocol { iso(1)member-body(2)us(840)
ieee802dot1partE(10010)asn1Module(2)loadprotocol(0)version1(0) }
                                    -- Imported from ISO/IEC 15802-4 : 1994

;  -- End of IMPORTS

LanManManagementPDU ::= CHOICE {
   loadPDU      [1]                 LoadPDU--Imported from ISO/IEC 15802-4 : 1994
   cpdu         [8]                 IMPLICIT CPDU }

-- NOTE: Implicit tags [0], and [2] through [7] were used by
-- PDUs defined in earlier drafts of ISO/IEC 15802-2. The use of these
-- PDU formats is no longer supported by ISO/IEC 15802-2
-- and is deprecated.
-- The use of other tag values is reserved.

-- The data type CPDU provides the ASN.1 definition of the CPDU structure
-- defined in table 7-5.

CPDU ::= SEQUENCE {
   fixedHeader  OCTETSTRING,         -- Fixed length octetstring, 20 octets long
   userInfo     UserData OPTIONAL }  -- Absent in CPDUs that are used purely for
                                     -- acknowledgments.

-- The data type UserData provides the ASN.1 definition of the User Info
-- field of the CPDU.

UserData ::= SEQUENCE {
   syntax       OBJECT IDENTIFIER,       -- Identifies the abstract syntax name for
                                         -- the contents of the user information fields

   version      BITSTRING                -- Carries the protocol version, encoded as a
                                         -- bitstring, as required by the protocol
                                         -- concerned.

   userPDU      ANY }                    -- Carries the protocol data unit. The value
                                         -- shall be encoded using a single ASN.1 type
                                         -- defined within the abstract syntax named by
                                         -- the "syntax" field above.
END
```

7.4 Use of underlying services by the CPE

The underlying service that is assumed by the CPE to provide for the transfer of CPDUs is unacknowledged connectionless-mode service, as defined in ISO/IEC 8802-2 : 1994, and provided by LLC Type 1 operation.

The use of other LLC Types of operation is not precluded. However, this International Standard does not specify any conformance-related aspects of the use of other LLC Types.

The minimum quality of service for the exchange of CPDUs is the minimum quality of service available at a local LSAP, using the service provided by LLC Type 1 operation.

7.4.1 Use of LLC Type 1 operation by the CPE

The use of the service provided by LLC Type 1 by the CPE shall be as follows:

The CPE generates a single DL-UNITDATA request primitive for each CPDU that the CPE requires to send. This request is mapped onto a DL-UNITDATA request as follows:

a) The source_address and destination_address in the DL-UNITDATA request primitive specify, respectively, the address of the sending CPE and the Destination Address parameter of the corresponding convergence service primitive;

b) The address fields of the resultant LLC PDU shall contain the standard LLC address reserved for use by the LAN/MAN Management protocol, as follows:

   The *address type designation* bit of the DSAP address shall be set to the bit pattern

   '0'

   The *actual address* bits of both the DSAP address and the SSAP address shall be encoded with the bit pattern

   '100 0000'

   where the leftmost bit is the least significant bit and the bits increase in significance from left to right.

   NOTE 1—The allocation of this standard LLC address for use by the LAN/MAN Management protocol is recorded in ISO/IEC TR 10178 : 1992. The use of this LLC address is a requirement for interoperability. The use of other LLC addresses, either as SSAP or as DSAP, is outside the scope of this International Standard.

c) The data parameter in the DL-UNITDATA request primitive shall contain a valid CPDU;

d) The priority parameter in the DL-UNITDATA request primitive shall indicate the Quality of Service indicated by the QOS parameter in the corresponding CPE-Data primitive.

DL-UNITDATA indication primitives generated by the local LLC subsystem that specify a source_address corresponding to the individual LLC address reserved for LAN/MAN Management and that carry valid instances of CPDUs shall be passed to the CPE.

NOTES

2—Where a station implements both the LAN/MAN Management protocol and the ISO/IEC 15802-4 : 1994 System load protocol, both LoadPDUs and CPDUs may be received on the LLC address reserved for LAN/MAN Management. Any LoadPDUs received are processed as described in ISO/IEC 15802-4. LoadPDUs and CPDUs are distinguished from each other by means of the ASN.1 tag values allocated to each PDU type, as defined in 7.3.6.3. The processing of each PDU type is independent of the processing of the other PDU type.

3—It should be borne in mind that the use of LLC Type 1 services may impose restrictions on maximum CPDU length, depending on the implementation of LLC and MAC in the local station.

4—The use of LLC Type 1 services involves the least processing in the communicating stations, and is intended for use in simple systems in LAN/MAN environments. As such, use of the LAN/MAN Management protocol is limited by the addressing capability of the link layer in LAN/MAN environments.

8. LAN/MAN Management managed object definitions

The management of the LMMP and the CPE is achieved via a number of managed objects. In addition, the access control information used by the Agent to determine the permitted access rights for all users of the LAN/MAN Management protocol is accessible via managed objects.

This clause defines the managed object classes associated with these management functions, along with their behaviour, attributes, operations, and notifications.

The managed object class definitions are expressed by use of the Template notation described in ISO/IEC 10165-4 : 1992. Further information on the use of this notation in the context of the LAN/MAN Management protocol can be found in ISO/IEC 10742 : 1994 and in IEEE Std 802.1F-1993. A set of ASN.1 encodings required by these managed object class definitions appears in 8.8.

8.1 Overview of managed object structure

The management information required for the management of a station using the LAN/MAN Management protocol falls into the following categories:

a) Station information
b) LMMPE/CPE information
c) Event notification routing information
d) Station access control information
e) Load protocol information

The definition of the LAN/MAN Management managed object classes describes the first four types of management information. The fifth type, Load protocol information, is described in ISO/IEC 15802-4 : 1994.

The following managed object classes are defined for the purposes of managing a LAN/MAN station; support of each managed object class is mandatory or optional as indicated:

— LAN/MAN Management managed object class (8.2—mandatory)
— Specific CPE info managed object class (8.3—optional)
— Resource Type ID managed object class (8.4—mandatory)
— Access Class Table Entry managed object class (8.5—optional)
— Notification Type Table Entry managed object class (8.6—optional)
— Event Report Destination Table Entry managed object class (8.7—optional)

The LAN/MAN Management managed object class is contained at the highest level within the containment hierarchy of a managed system; i.e., it is contained within an instance of the system managed object class. The definition of the system managed object class is to be found in SMI Part 2 (ISO/IEC 10165-2 : 1992). There is a one-to-one relationship between instances of the LAN/MAN Management managed object class and instances of the system managed object class.

A single instance of the Resource Type ID managed object class is contained within the LAN/MAN Management managed object class.

Multiple instances of the Access Class Table Entry, Notification Type Table Entry, and Event Report Destination Table Entry managed object classes may be contained within each instance of the LAN/MAN Management managed object class.

The following subclauses define these managed object classes, their behaviour, attributes, operations, and notifications.

8.2 LAN/MAN Management managed object class definition

This managed object class provides functionality that supports the management of CPE timer and retry counter information.

Support for this managed object class is mandatory.

```
oLANMANManagement  MANAGED OBJECT CLASS
   DERIVED FROM "ISO/IEC 10165-2 : 1992":top;
   CHARACTERIZED BY
      pLANMANMgt    PACKAGE
         ATTRIBUTES     aLMMName         GET,             -- Naming Attribute
                        aDefaultCPEInfo  GET-REPLACE;
      ;
   ;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007) managedObjectClass(3)
               lmmclass(0)};

nbLMMBinding NAME BINDING
   SUBORDINATE OBJECT CLASS          oLANMANManagement
                AND SUBCLASSES
   NAMED BY SUPERIOR OBJECT CLASS    "ISO/IEC 10165-2 : 1992":system
                                     AND SUBCLASSES;
   WITH ATTRIBUTE                    aLMMName;
   BEHAVIOUR
      bLMMBinding       BEHAVIOUR
         DEFINED AS     !A single instance of the LAN/MAN Management managed object class
                        exists within the superior object class. It cannot be created or
                        deleted dynamically by management action!;
   ;
REGISTERED AS  {iso(1)member-body(2)us(840)ieee802-1B(10007)nameBinding(6)lmmbinding(0)};
```

8.2.1 LMM Name attribute

```
aLMMName ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  IEEE802-1-LMMDefinitions.LMMName;
   MATCHES FOR  EQUALITY;
   BEHAVIOUR
      bLMName           BEHAVIOUR
         DEFINED AS     !This attribute is used to name the instance of the LAN/MAN Management
                        managed object within the systems managed object. The value of this
                        name attribute is fixed and is equal to the string "LMM".!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) lmmname(0)};
```

8.2.2 Default CPE Info attribute

```
aDefaultCPEInfo ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  IEEE802-1-LMMDefinitions.DefaultCPEInfo;
   MATCHES FOR EQUALITY;
   BEHAVIOUR
      bDefaultCPEInfo   BEHAVIOUR
                        DEFINED AS   !This attribute defines the default values to be used
                                     for the establishment of timeout values and retry
                                     limits for CPE requests.  The attribute is a single-
                                     valued type, with the value structured as a pair of
                                     fields.  One field carries an integer corresponding
                                     to the retry limit, the second field carries a real
                                     value representing the timeout period in units of
                                     seconds.  In the absence of specific values related
                                     to identified affiliates, these values are used to
                                     set the corresponding fields in the affiliate
                                     record.!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7)
              defaultCPEinfo(1)};
```

8.3 Specific CPE Info managed object class

This managed object class defines specific values to be used for the establishment of timeout values and retry limits for CPE requests to a specific CPE address. Zero or more instances of this managed object class may be contained within the LAN/MAN Management managed object class; each instance defines values related to a different CPE address. The values defined in these managed objects override the default values contained in the Default CPE Info attribute in the LAN/MAN Management managed object for the specific CPE address to which they relate.

Support for this managed object class is optional.

```
oSpecificCPEInfo MANAGED OBJECT CLASS
   DERIVED FROM "ISO/IEC 10165-2 : 1992":top;
   CHARACTERIZED BY
      pSpecificCPEInfo PACKAGE
         BEHAVIOUR
            bSpecificCPEInfo BEHAVIOUR
               DEFINED AS  !This managed object class defines, for CPE requests destined
                           for the CPE reachable at the address contained in the CPE
                           Address attribute, the values to be used for the
                           establishment of retry limits and timeout values.!;
         ;
         ATTRIBUTES   aCPEAddress GET,     -- Naming attribute
                      aDefaultCPEInfo GET-REPLACE
         ;
      ;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) managedObjectClass(3)
               specificCPEinfo(1)};

nbSpecificCPEInfo NAME BINDING
   SUBORDINATE OBJECT CLASS oSpecificCPEInfo AND SUBCLASSES;
   NAMED BY
   SUPERIOR OBJECT CLASS oLANMANManagement AND SUBCLASSES;
   WITH ATTRIBUTE aCPEAddress;
   CREATE;
   DELETE;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) nameBinding(6)
               specificCPEbinding(1)};
```

8.3.1 CPE Address attribute

```
aCPEAddress ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-LMMDefinitions.LSAPAddress;
   MATCHES FOR EQUALITY;
   BEHAVIOUR
      bCPEAddress BEHAVIOUR
         DEFINED AS   !This attribute carries the CPE address of a remote CPE. The CPE
                      address is the LSAP address of the CPE.!
      ;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) cpeaddress(2)};
```

8.4 Resource Type ID managed object class

A single instance of the Resource Type ID managed object class is contained within the LAN/MAN Management managed object class. The managed object class definition itself is contained in ISO/IEC 10742 : 1994, therefore only the name binding appears in this International Standard; however, implementation of the managed object in accordance with the definition contained in ISO/IEC 10742 is a conformance requirement of this International Standard. The Resource Type ID managed object class contains manufacturer and product information related to the implementation of the management functionality.

Support for this managed object class is mandatory.

```
nbResourceIDBinding   NAME BINDING
   SUBORDINATE OBJECT CLASS       "ISO/IEC 10742 : 1994":oResourceTypeID AND SUBCLASSES;
   NAMED BY SUPERIOR OBJECT CLASS  oLANMANManagement AND SUBCLASSES;
   WITH ATTRIBUTE                  "ISO/IEC 10742 : 1994":aResourceTypeIDName;
   BEHAVIOUR
      bResourceIDBinding     BEHAVIOUR
         DEFINED AS   !A single instance of the Resource Type ID managed object class
                      exists within the superior object class. It cannot be created or
                      deleted dynamically by management action.!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) nameBinding(6)
              resIDbinding(2)};
```

8.5 Access class table entry managed object class definition

The information necessary to properly verify access rights for a request is contained in the Access Class Table. Each entry (“row”) in the table defines, for a particular managed object class/password combination, what access classes are available for local or remote access. The managed object classes that appear in the table may be any class that is accessible by the LAN/MAN Management protocol in the station. Each entry contains a managed object class, a password, a set of access classes for access by local managers, and a set of access classes for access by remote managers. The access class required for access to each attribute of a managed object is a local matter.

This managed object class represents a single entry (“row”) in the Access Class Table; it therefore permits individual table entries to be manipulated independently. An instance of this class exists for each entry in the table.

Support for this managed object class is optional.

```
oAccessClassTableEntry  MANAGED OBJECT CLASS
   DERIVED FROM    "ISO/IEC 10165-2 : 1992":top;
   CHARACTERIZED BY
      pAccessClassTableEntry     PACKAGE
         ATTRIBUTES              aAccessClassTableEntryName   GET,
                                 aManagedObjectClasses        GET-REPLACE,
                                 aPassword                    GET-REPLACE,
                                 aLocalAccessClasses          GET-REPLACE,
                                 aRemoteAccessClasses         GET-REPLACE;
      ;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) managedObjectClass(3)
              accessentry(2)};

nbAccessClassTableEntry  NAME BINDING
   SUBORDINATE OBJECT CLASS          oAccessClassTableEntry AND SUBCLASSES;
   NAMED BY SUPERIOR OBJECT CLASS    oLANMANManagement AND SUBCLASSES;
   WITH ATTRIBUTE                    aAccessClassTableEntryName;
   BEHAVIOUR
      bAccessEntryBinding     BEHAVIOUR
      DEFINED AS              !A single instance of the Access Class Table Entry managed
                              object class exists in the LAN/MAN Management managed object
                              for each entry that exists in the Access Class Table, as
                              described above.!;
   ;
   CREATE;
   DELETE;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) nameBinding(6)
              atebinding(3)};
```

8.5.1 Access class table entry name attribute

```
aAccessClassTableEntryName  ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-LMMDefinitions.EntryIndex;
   MATCHES FOR  EQUALITY;
```

```
   BEHAVIOUR
      bAccessClassTableEntryName   BEHAVIOUR
         DEFINED AS      !This attribute is used to name the instances of the Access Class
                         Table Entry managed object within the LAN/MAN Management managed
                         object. The value of this name attribute is an integer value and is
                         equal to the entry number in the Access Class Table.!;
   ;
REGISTERED AS   {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) atename(3)};
```

8.5.2 Managed object classes attribute

```
aManagedObjectClasses  ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  IEEE802-1-LMMDefinitions.ManagedObjectClasses;
   MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
   BEHAVIOUR
      bManagedObjectClasses  BEHAVIOUR
         DEFINED AS      !This attribute carries the value of the set of managed object
                         classes to which the remaining information in the table entry
                         managed object relates.  If the set has no members, the value is
                         interpreted as meaning "all managed object classes".!;
   ;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) moclass(4)};
```

8.5.3 Password attribute

```
aPassword ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-LMMDefinitions.Password;
   MATCHES FOR EQUALITY, SUBSTRINGS;
   BEHAVIOUR
      bPassword BEHAVIOUR
         DEFINED AS      !This attribute carries the value of the password that is
                         required to be provided along with a local or remote operation
                         request on the managed object class(es) to which the table entry
                         applies, in order for the local or remote access classes to be
                         granted.!;
   ;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) password(5)};
```

8.5.4 Local access classes attribute

```
aLocalAccessClasses ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-LMMDefinitions.LocalAccessClasses;
   MATCHES FOR EQUALITY;
   BEHAVIOUR
      bLocalAccessClasses  BEHAVIOUR
         DEFINED AS      !This attribute carries a bitstring representing the value of the
                         set of access classes that are available if the correct password
                         is provided with a local operation on managed objects of any
                         class to which this table entry applies.!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7)
               localclasses(6)};
```

8.5.5 Remote access classes attribute

```
aRemoteAccessClasses ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-LMMDefinitions.RemoteAccessClasses;
   MATCHES FOR EQUALITY;
   BEHAVIOUR
      bRemotelAccessClasses BEHAVIOUR
         DEFINED AS      !This attribute carries a bitstring representing the value of the
                         set of access classes that are available if the correct password
                         is provided with a remote operation on managed objects of any
                         class to which this table entry applies.!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7)
               remoteclasses(7)};
```

8.6 Notification type table entry managed object class definition

The information necessary to properly route notification information that is passed to the agent from a particular managed object is organized into two data tables, the Notification Type Table and the Event Report Destination Table. The Notification Type Table entries define, for a combination of managed object class(es) and notification type(s), the event report destination(s) to which events of that class from the specified managed object class(es) are to be delivered. A Notification Type Table Entry managed object exists for each entry in the table. The notification type is an object identifier value, as allocated in the definition of the corresponding notification. Each table entry contains a set of managed object classes, a set of notification types, and a set of Event Report Destination Indexes. An Event Report Destination Index is an integer value that identifies an entry in the Event Report Destination Table, which provides addressing information for delivering event reports. A value of the empty set for the set of Event Report Destination Indexes indicates the "null" destination, that is, that no event report is desired for that set of notification types from that set of managed object classes.

This managed object class represents a single entry ("row") in the Notification Type Table; it therefore permits individual table entries to be manipulated independently. An instance of this class exists for each entry in the table.

Support for this managed object class is optional.

```
oNotificationTypeTableEntry  MANAGED OBJECT CLASS
   DERIVED FROM    "ISO/IEC 10165-2 : 1992":top;
   CHARACTERIZED BY
      pNotificationTypeTableEntry  PACKAGE
         ATTRIBUTES     aNotificationTypeTableEntryName      GET,
                        aManagedObjectClasses                GET-REPLACE ADD-REMOVE,
                        aNotificationTypes                   GET-REPLACE ADD-REMOVE,
                        aEventReportDestinations             GET-REPLACE ADD-REMOVE;
      ;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) managedObjectClass(3)
              notificationentry(3)};

nbNotificationTypeTableEntry  NAME BINDING
   SUBORDINATE OBJECT CLASS          oNotificationTypeTableEntry AND SUBCLASSES;
   NAMED BY SUPERIOR OBJECT CLASS    oLANMANManagement AND SUBCLASSES;
   WITH ATTRIBUTE                    aNotificationTypeTableEntryName;
   BEHAVIOUR
      bNotificationTypeTableEntryBinding  BEHAVIOUR
         DEFINED AS     !A single instance of the Notification Type Table Entry managed
                        object class exists in the LAN/MAN Management managed object for
                        each entry that exists in the Notification Type Table, as
                        described above.!;
   ;
   CREATE ;
   DELETE;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) nameBinding(6)
              notificationentrybinding(4)};
```

8.6.1 Notification type table entry name attribute

```
aNotificationTypeTableEntryName  ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  IEEE802-1-LMMDefinitions.EntryIndex;
   MATCHES FOR         EQUALITY;
   BEHAVIOUR
      bNotificationTypeTableEntryName  BEHAVIOUR
         DEFINED AS     !This attribute is used to name the instances of the Notification
                        Type Table Entry managed object within the LAN/MAN Management
                        managed object. The value of this name attribute is an integer
                        value and is equal to the entry number in the Notification Type
                        Table.!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7)
              notypename(8)};
```

8.6.2 Notification types attribute

```
aNotificationTypes ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-LMMDefinitions.NotificationTypes;
   MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
   BEHAVIOUR
      bNotificationTypes BEHAVIOUR
         DEFINED AS     !This attribute carries the set of notification types for which
                        the table entry managed object applies. An attribute value that
                        contains no notification types (the "empty set") is interpreted
                        as meaning "all notification types".!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) notypes(9)};
```

8.6.3 Event report destinations attribute

```
aEventReportDestinations ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-LMMDefinitions.EventReportDestinations;
   MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
   BEHAVIOUR
      bEventReportDestinations BEHAVIOUR
         DEFINED AS     !This attribute carries the index values of the set of event
                        report destination table entries to be used to provide
                        destination addresses for event reports resulting from
                        notifications that match the notification types and managed
                        object class specified in the notification type table entry. If
                        the attribute contains no index values, no event reports are
                        generated for that combination of managed object class and
                        notification types!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7)
              evendests(10)};
```

8.7 Event report destination table entry managed object class definition

The Event Report Destination Table contains one entry for each possible destination for event reports, and there is an Event Report Destination Table entry managed object that corresponds to each entry. Each entry contains the value of an Event Report Destination Index and a destination CPE address. A value of zero for the CPE address in an entry indicates that the local manager is the destination. Optionally, a QOS value may be associated with each destination, indicating the quality of service to be used when delivering event reports to that destination.

This managed object class represents a single entry (“row”) in the Event Report Destination Table; it therefore permits individual entries to be manipulated independently. An instance of this class exists for each entry in the table.

Support for this managed object class is optional.

```
oEventReportDestinationTableEntry  MANAGED OBJECT CLASS
   DERIVED FROM "ISO/IEC 10165-2 : 1992":top;
   CHARACTERIZED BY
      pEventReportDestinationTableEntry   PACKAGE
         ATTRIBUTES     aEventReportDestinationTableEntryName    GET,
                        aEventReportDestinationAddress           GET-REPLACE;
      ;
   ;
   CONDITIONAL PACKAGES
      pQOSPackage PACKAGE
         ATTRIBUTES          aDestinationQOS GET-REPLACE;
      REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007)
                     packages(4) qospackage(0)};
   PRESENT IF !enhanced Reliability QOS supported!;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) managedObjectClass(3)
              eventdestentry(4)};
```

```
nbEventReportDestinationTableEntry  NAME BINDING
   SUBORDINATE OBJECT CLASS        oEventReportDestinationTableEntry AND SUBCLASSES;
   NAMED BY SUPERIOR OBJECT CLASS  oLANMANManagement AND SUBCLASSES;
   WITH ATTRIBUTE                  aEventReportDestinationTableEntryName;
   BEHAVIOUR
      bEventReportDestEntryBinding BEHAVIOUR
         DEFINED AS      !A single instance of the Event Report Destination Table Entry
                         managed object class exists in the LAN/MAN Management managed
                         object for each entry that exists in the Event Report Destination
                         Table.!;
   ;
   CREATE;
   DELETE;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) nameBinding(6)
              eventdestentrybinding(5)};
```

8.7.1 Event report destination table entry name attribute

```
aEventReportDestinationTableEntryName  ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  IEEE802-1-LMMDefinitions.EntryIndex;
   MATCHES FOR         EQUALITY;
   BEHAVIOUR
      bEventReportDestinationTableEntryName  BEHAVIOUR
         DEFINED AS      !This attribute is used to name the instances of the Event Report
                         Destination Table Entry managed object within the LAN/MAN Management
                         managed object. The value of this name attribute is an integer value
                         and is equal to the entry number in the Event Report Destination
                         Table.!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) edename(11)};
```

8.7.2 Event report destination address attribute

```
aEventReportDestinationAddress  ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  IEEE802-1-LMMDefinitions.EventReportDestinationAddress;
   MATCHES FOR         EQUALITY;
   BEHAVIOUR
      bEventReportDestinationAddress  BEHAVIOUR
         DEFINED AS      !This attribute specifies the destination address to be used when
                         sending event reports. Its value may be Null, indicating local
                         delivery only, or may be equal to a CPE address.!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) ede(12)};
```

8.7.3 Destination QOS attribute

```
aDestinationQOS ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-LMMDefinitions.DestinationQOS;
   MATCHES FOR EQUALITY;
   BEHAVIOUR
      bDestinationQOS    BEHAVIOUR
         DEFINED AS      !This attribute specifies the QOS value to be used when sending
                         event reports to a given CPE address, as specified by the CPE
                         address attribute.!;
   ;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7) qos(13)};
```

8.8 Data definitions for the LMM managed objects

This subclause defines data elements that are associated with the managed object class definitions and that may be conveyed in management protocol.

8.8.1 Notation for the data description

The description consists of an ASN.1 module that describes the structure of data elements (attribute values, information syntaxes for notifications, etc.) that are specific to the LMM managed object classes.

It should be noted that the description of these elements does not represent a constraint on how the data is stored or manipulated within a given implementation; it is merely a constraint on the representation that shall be used when elements of the managed objects are described within a management protocol data unit as part of a management protocol exchange.

8.8.2 LMM ASN.1 module

```
IEEE802-1-LMMDefinitions {iso(1) member-body(2) us(840) ieee802-1B(10007) asn1Module(2)
lmmdefinitions(1) version1(0)}
DEFINITIONS ::=
BEGIN

IMPORTS

MACAddress
FROM IEEE802CommonDefinitions {iso(1) member-body(2) us(840) ieee802dot1partF(10011)
asn1Module(2) commondefinitions(0) version1(0)} --Imported from IEEE Std 802.1 PartF

;  -- End of IMPORTS

-- Syntax of LAN/MAN Management Managed Object Naming Attribute

LMMName ::= GraphicString {"LMM"}

-- Syntax of Default CPE Info

DefaultCPEInfo ::= SEQUENCE {
             defaultRetryLimit   INTEGER,
             defaultTimeout      REAL }
-- Note: The encoding used for the defaultTimeout REAL value shall be restricted to
-- use only the NR2 form, as defined in ISO 6093 : 1985. The use of this encoding within
-- the Basic Encoding Rules is defined in ISO/IEC 8825 : 1990.


-- Syntax of Managed Object Classes attribute

ManagedObjectClasses ::= SET OF OBJECT IDENTIFIER

-- Syntax of Password attribute

Password ::= GraphicString

-- Syntax of Local Access Classes attribute

LocalAccessClasses ::= BITSTRING

-- Syntax of Remote Access Classes attribute

RemoteAccessClasses ::= BITSTRING

-- Syntax of Notification Types attribute

NotificationTypes ::= SET OF OBJECT IDENTIFIER

-- Syntax of EventReportDestinations attribute

EventReportDestinations ::= SET OF EntryIndex

-- Syntax of Event Report Destination Address attribute

EventReportDestinationAddress ::= CHOICE {
   localOnly    NULL,
   cpeAddress   LSAPAddress }

-- Syntax of LSAP address
LSAPAddress ::= SEQUENCE {
       llcAddress OCTETSTRING,
       macAddress MACAddress }
```

```
-- An LSAP address consists of an LLC address, concatenated with a MAC address.
-- The LLC address consists of a single octet that represents the LSAP number,
-- followed by a further 5 octets for the Protocol Identifier in the case where the LSAP number
-- is the number of the SNAP SAP, as defined in IEEE Std 802-1990.
-- The LSAP number assigned to the SNAP SAP is recorded in ISO/IEC TR 10178 : 1992.
-- The least significant bit (the address type designation bit) of the LSAP
-- number is encoded as the least significant bit of the first octet of llcAddress.
-- The octet encoding for the Protocol Identifier, if present, is derived from the
-- hexadecimal display representation order of the identifier, AB-CD-EF-GH-IJ, where
-- AB-CD-EF is the representation of the Organizationally Unique Identifier. The
-- octets are encoded as follows:  The first pair of hexadecimal digits are
-- encoded in the second octet of llcAddress, the second pair in the third octet,
-- and so on.

-- Syntax of Destination QOS attribute

DestinationQOS ::= INTEGER {
   basicReliability (0),
   enhancedReliability (1) }

-- Syntax of Entry Index

EntryIndex   ::= INTEGER
END
```

9. Event forwarding and access control

This clause describes the functionality of the Agent that is controllable by means of the Access Class Table Entry, Notification Type Table Entry, and Event Destination Table Entry managed objects.

9.1 Event forwarding

The event-forwarding mechanisms described in this International Standard provide a simple means whereby a notification emitted by a managed object can be forwarded to one or more Managers in the form of an event report. The forwarding information is contained in two tables—the *notification type table* and the *event report destination table*.

Each entry in the event report destination table defines a destination CPE address to which event reports may be delivered, and may also indicate the quality of service to be used for delivery to that destination. A value of null for the destination CPE address indicates that the destination is the local manager.

Each entry in the notification type table contains a set of OBJECT IDENTIFIERs under which managed object classes have been registered, a set of OBJECT IDENTIFIERs under which notifications have been registered (the notification types), and a set of pointers to entries in the event report destination table.

When a notification is emitted by a managed object, the Agent searches the notification type table for an entry that contains the notification type corresponding to the notification emitted and the managed object class corresponding to the managed object that emitted it. If no such entry exists, the notification is ignored. If an entry exists, the Agent forwards the notification, in the form of an event report, to each destination CPE address indicated by the set of pointers to event report destination table entries. If there are no pointers, the notification is ignored. Figure 9-1 illustrates the table structure.

In the example, notifications of type P or Q emitted by managed objects of class A or C will not result in event reports, whereas notifications of type P or Q emitted by managed objects of class B or D will be delivered as event reports to destination addresses V, X, and Z; notifications of type S, T, or U from managed objects of class B, D, or E will be delivered as event reports to destination address W, etc. Event reports will be delivered to destinations X and Z with Enhanced QOS; all others will be delivered with Basic QOS.

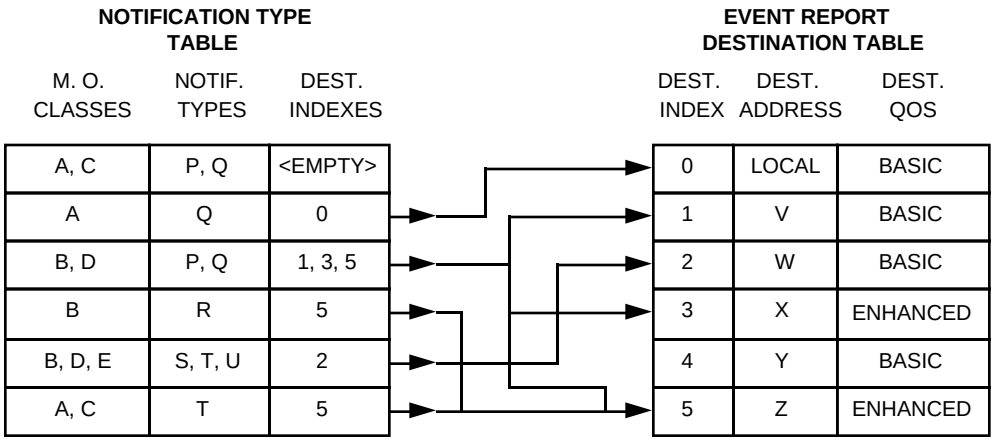


Figure 9-1—Notification type and event report destination tables

9.2 Access control

The access control mechanisms defined in this International Standard are intended to provide a means to allow or deny access to managed objects within a station. These mechanisms are designed to prevent accidental or incorrect use of management operations and casually malicious abuse of management operations. The mechanisms alone are not intended to implement a secure environment; thus the requirements for such an environment are not considered here. Authentication, confidentiality, and integrity of management traffic may be provided by the use of the mechanisms described in IEEE Std 802.10-1992.

The access control mechanism described in this clause provides the following functionality:

a) Enabling and disabling of access control for individual managed object classes.
b) Provision of access control information that may be used to control access to individual attributes.
c) Access control based on the source of the request. The CPE addresses of both parties involved in a request are available to the access control mechanism.

Implementation-specific restrictions on access to managed objects, based on other factors, are in no way precluded or prohibited by the provision of International Standard access control mechanisms. Equally, no particular implementation model is mandated or implied by this access control mechanism.

9.2.1 General

Access control is based on the concept of access classes. In the local system, each operation and managed object/attribute combination may have associated with it a property determining which access classes possess sufficient privilege to allow that operation to be performed; the manner in which these access classes are allocated is a local matter.

A remote Manager that wishes to gain access to a managed object includes a “password” and a desired set of access classes with each protected operation that it wishes to perform; this information is passed in the access control fields of the LMMP. The local Agent makes use of the password information to determine what set of access classes is permissible for the remote manager; if the set of classes requested by the remote manager is a subset of those allowable, the requested set of access classes is granted to the remote manager. Any requests to access managed objects are now associated with the granted set of access classes.

NOTE—The access control mechanisms described here allow a similar mechanism to be applied to requests originating from a local LMMU. As interactions between a local LMMU and a locally accessible managed object involve no exter-

nally visible protocol, the only aspect of the control of such local operations that is specified by this International Standard is the means whereby a remote Manager may determine and modify the access rights available to the local LMMU.

The mapping of access control information to access classes is common to all managed object classes, and the control function is shown in figure 9-2. This function is carried out in addition to the service and protocol definitions described in clauses 6 and 7.

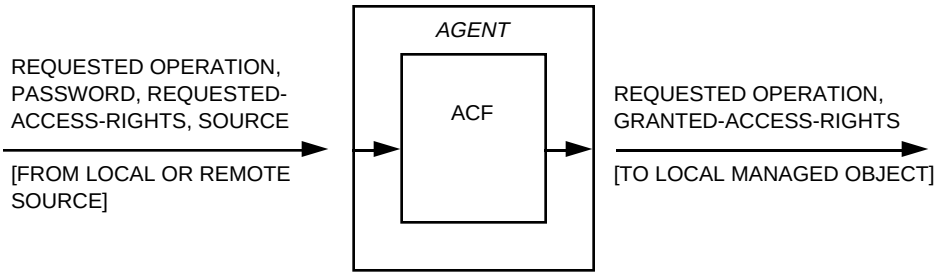


Figure 9-2—Access control function

9.2.2 Access control function

Access control information is contained in the Access Class Table Entry managed objects. The AccessClassTable contains access control information relating to each set of managed object classes for which access control is required; for each set of managed object classes, a number of passwords may be defined, each of which has associated with it a Local Access Class and a Remote Access Class attribute. These attributes define the set of permitted Access Classes for Remote or Local requests that specify that password/managed object class combination. The AccessClassTable is referenced when a request for an operation is made, in order to determine what access class is to be permitted for that operation on a given managed object class. The table entry managed objects are defined in 8.5. The Agent maps the received access control information to the local or remote access classes as appropriate using the algorithm described in 9.2.3, and the permitted access classes are then associated with the resultant operation request. The Agent allows or disallows the operation on the basis of a comparison between the permitted access classes and the access classes that are required for the operation concerned. The sets of required access classes for reading and/or writing to each attribute are defined on a per-implementation basis.

The AccessClassTable contains zero or more entries. Each entry provides a mapping from a password octetstring and local/remote source information onto a permitted set of access classes.

The permitted set of access classes is encoded as a bitstring, where each bit is used to encode the granting or denying of a particular access class. The allocation of meanings to these classes is a local matter.

9.2.3 Agent algorithm

An Agent shall follow the following procedure upon receipt of a request for an operation where Access Control is implemented in the local station. The procedure shall be carried out on a per-operation basis.

Access Control information may be received in either the access control field of an LMMS primitive, or in the case of locally originated requests, via a locally defined mechanism. If no Access Control information is supplied to the Agent, the access_classes associated with the subsequent operation request to the managed object shall be null.

If Access Control information is included, then, firstly, the password field of access control table entries relating to the managed object that is being accessed is searched for an entry that exactly matches the supplied password field. The test shall fail if

a) No match for the password is found, or
b) An AccessControl field containing no password is supplied, or
c) A password match is found, but the set of desired access classes in the Access Control Information does not equal or subset the set of access classes associated with the password for local or remote access (as appropriate).

If the test fails, the operation is rejected. If the test succeeds, the desired set of access classes is associated with the subsequent operation request to the managed object.

The Agent is responsible for acting on the granted access class information associated with the operation request. For each operation that may be performed upon a managed object of a given class, the local system may define a set of access classes that represent the minimum requirement in order for permission to be granted to perform that operation. This definition is a local matter. A comparison is performed between the granted and required sets of access classes; for the operation to be performed, the permitted set shall equal or superset the required set. If no access classes have been defined for an operation on a managed object class, the value “null” shall constitute the set of required classes for that operation. If the permitted access classes are insufficient for the operation concerned, the operation shall be rejected.

10. Conformance

10.1 Static conformance

10.1.1 LMMP static conformance

An implementation for which conformance to this International Standard is claimed shall conform to the static conformance requirements of CMIP (ISO/IEC 9596-1 : 1991), excluding those that relate to the ACSE protocol and associations.

10.1.2 Convergence protocol static conformance

An implementation for which conformance to this International Standard is claimed shall support the following:

a) The use of the Convergence Protocol by LMMP as specified in 7.2
b) The following CPE capabilities, as specified in 7.3:
    1) The ability to receive unconfirmed requests
    2) The ability to receive and respond to confirmed requests
    3) The ability to generate unconfirmed requests
    4) The ability to generate confirmed requests and receive their responses
    5) The encoding and decoding of CPDUs, including the LMM_PDUs conveyed in User Info fields, in accordance with the requirements specified in 7.3.6.2 and the Basic Encoding Rules specified in ISO/IEC 8825 : 1990
c) The use of LLC Type 1 procedures for transfer of CPDUs as specified in 7.4

10.1.3 Managed objects

An implementation for which conformance to this International Standard is claimed shall support

a) Any managed object classes specified as mandatory in clause 8
b) For each managed object class supported, all the associated attributes, operations, notifications, behaviour, and name bindings

Support for any element of management information and any associated ASN.1 constructions defined in IEEE Std 802.1F-1993 is only required when such information is referenced by a managed object class definition for which support is claimed.

NOTE—Support for other managed object classes is outside the scope of this International Standard; in general, such support will be required, or optional, as specified in other standards that define managed object classes and management information types to which conformance is claimed and that are implemented in the same system as LMMP.

10.1.4 Options

For an LMMP implementation, support of the CMISE multiple object selection, filter, multiple reply, and cancel get functional units is optional, as in CMIP (ISO/IEC 9596-1 : 1991).

For a Convergence Protocol implementation, support of multiple outstanding requests is optional; i.e., the maximum group size supported may be one.

For managed object classes, support is optional as defined in clauses 8, 11.5, and 11.6.1.

For an implementation claiming support of the DEFED protocol, support of the protocol specified in 11.4 is required.

10.2 Protocol implementation conformance statement

The supplier of a protocol implementation that is claimed to conform to this International Standard shall complete

a) A copy of the PICS proforma provided in annex A
b) A copy of the PICS proforma for CMIP, as defined in ISO/IEC 9596-2

These shall include in each case the information necessary to identify both the supplier and the implementation.

10.3 Dynamic conformance

The dynamic conformance requirements that follow apply to all LLC PDUs conveying LLC user data and transmitted or received via the LAN/MAN Management LSAP identified by the LLC *actual address* '100 0000'.

An implementation for which conformance to this International Standard is claimed shall exhibit external behaviour consistent with having implemented, for each function that either PICS states to be supported

a) The corresponding procedures of LMMP and the Convergence Protocol as specified in clause 7;
b) Where applicable, the mechanisms for event forwarding and access control specified in clause 9;
c) Where applicable, the mechanisms for discovery and event forwarding enable/disable specified in clause 11.

All transmitted PDUs shall be encoded as specified in 7.2.5 and 7.3.6.

All managed object specific information contained in PDUs transmitted as responses or event reports shall accurately reflect the instantaneous state or attribute values of the managed objects to which it relates.

11. Discovery and dynamic control of event forwarding

Management of stations within a LAN or MAN is generally based on responsibilities assigned to management stations with respect to managed objects within agent stations. As stations in a network can appear and disappear over time (for example, as a result of physical relocation, power down, and power up), and managed objects associated with stations can be dynamically created and deleted, the set of stations and managed objects that fall within the scope of a given manager's responsibility can also change over time.

This International Standard defines services and protocols for management that can be used in circumstances where manager and agent stations have *a priori* knowledge of each other's existence, responsibilities, and capabilities, and that can also be used in a number of ways to establish that knowledge in a dynamic manner. This clause describes a standardized mechanism that permits manager and agent stations to dynamically

a) Announce and/or discover each other's existence in the network.
b) Exchange information with respect to managed objects that are available to be managed in an agent station and that fall within the scope of a manager's responsibility.
c) Establish the correct information in, or remove obsolete information from, an agent station's Notification Type table and Event Report destination table.

The mechanism described makes use of the LAN/MAN Management Service (LMMS) defined in clause 6 and represents an optional addition to the functionality described in clauses 6–10. Other mechanisms for establishing and manipulating management knowledge and event forwarding information can be used in addition to or in place of the ones described here; however, the availability and use of a standardized mechanism will promote intervendor interoperability.

11.1 Scope

This subclause describes a mechanism that permits the dynamic discovery of manager and agent stations within a LAN/MAN environment, the exchange of management knowledge information between manager and agent stations, and the manipulation of notification type and event destination table information in agent stations. To this end, it

a) Describes an architecture for discovery and dynamic control of event forwarding.
b) Defines services that enable manager and agent stations to establish shared management knowledge (as defined in 5.4.4 of ISO/IEC 10040 : 1992) and to manipulate notification type table and event destination table entries.
c) Specifies protocol elements that provide those services.
d) Defines managed object classes and associated management information that support the provision and management of those services.

11.2 Architecture

11.2.1 Requirements addressed

The LMMS provides service primitives and managed object classes that allow Managers to monitor and control managed objects in Agents, and for Agents to send event reports to Managers. In a statically configured network, the information required by Manager and Agent stations in order for one to manage or be managed by the other can be established by *a priori* mechanisms, the details of which are outside the scope of standardization. In a dynamically configured network, where stations may come and go over time, and where the managed objects that they contain may be created and deleted dynamically, it is necessary to provide mechanisms that allow the details of any reconfiguration to be communicated between Manager and Agent stations. Specifically, Manager stations require

a) Knowledge of the existence of Agent stations in the network that may fall within the scope of their management responsibility.
b) Knowledge of the set of managed object instances that such stations contain.
c) Knowledge of changes resulting from the appearance or disappearance of Agent stations, or their managed object instances.
d) Mechanisms that allow them to establish or remove event report forwarding information in Agent stations having gained such knowledge.

Agent stations require

— The ability to advertise the managed objects that they support to Manager stations in the network that may wish to manage them.
— The ability to respond to requests for management knowledge from Managers that may wish to manage them.
— Mechanisms that allow them to establish event report forwarding information at the request of a Manager, or to remove event report forwarding information either at the request of a Manager or as the result of a unilateral decision on the part of the Agent.

11.2.2 The model

The requirements described in 11.2.1 are modelled as mechanisms that fall into two categories:

a) Discovery mechanisms
b) Event forwarding enable/disable mechanisms

These mechanisms are modelled as an initial information exchange, followed for some mechanisms by one, or a set of, responding information exchanges. Each exchange occurs between an originating station, which can be a Manager or an Agent station, and a set of receiving stations, which are Agent stations or Manager stations, respectively. The information exchanges are defined in terms of service primitives, for the *discovery service* (11.3.1) and the *event forwarding enable/disable service* (11.3.2). These services are provided by means of the protocol specified in 11.4. The exchanges involved in the event forwarding enable/disable mechanisms are associated with changes to the contents of the notification type and event report destination tables (see 8.6 and 8.7) in the Agent system(s) that are participating in the exchanges.

The set of receiving stations in an information exchange can be either a group of stations, identified in the LAN/MAN environment by a group MAC address, or a single individual station, identified in the LAN/MAN environment by an individual MAC address (see IEEE Std 802-1990). For each mechanism, the choice of addressing is restricted, as described below.

NOTES

1—It is possible that a group of stations could be empty at the time when an exchange is initiated; for example, if no Manager station is active when an Agent station originates an exchange addressed to the Manager's group.

2—In figures 11-1 through 11-5, the arrows between Manager and Agent boxes indicate the direction of flow of information between originating and receiving stations. Where the set of receiving stations (Manager or Agent) is represented by multiple boxes, the information exchange can use either individual or group addressing; otherwise, it uses individual addressing. Where the originating station is represented by multiple boxes, the receiving station can receive information from a number of originating stations belonging to a group (this occurs only for a responding exchange that follows an initial group addressed exchange; see figure 11-2).

11.2.2.1 Discovery

Discovery can be initiated by an Agent, either to announce the fact that the station that contains it has been installed in the network (for example, as a result of station power-on or connection of a new station) or to announce the instantiation of a managed object or objects. In both cases, the information provided by the Agent is directed either to an individual Manager station or to a group of Manager stations, and can include

configuration information detailing the managed object instances that the Agent station currently supports. Figure 11-1 illustrates this situation.

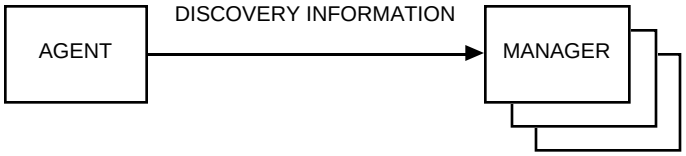


Figure 11-1—Agent-initiated discovery

Discovery can be initiated by a Manager when it is installed in the network (for example, as a result of station power-on or connection of a new station) in order to establish its initial view of the network, or at a later stage to verify that its current view of the state of the network is valid. In both cases, the information provided by the Manager is directed either to an individual Agent station or to a group of Agent stations, and can include information detailing the managed object classes and/or instances that the Manager station is interested in managing. Manager-initiated discovery also prompts the Agent station(s) to respond by giving information on their configuration, at the Agent's discretion. Figure 11-2 illustrates this situation.

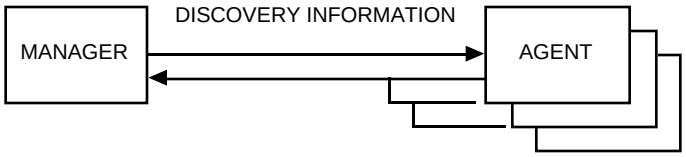


Figure 11-2—Manager-initiated discovery

11.2.2.2 Event forwarding enable/disable

Event forwarding enable and disable exchanges can be initiated by a Manager, either to cause event reports from particular Agent stations to be sent to it, or to stop event reports from being sent. In the case of enable exchanges, the information provided by the Manager is directed to a single Agent station; disable exchanges can be directed either to an individual Agent station or to a group of Agent stations. In all cases, the information can detail the managed object instances whose notifications are the subject of the exchange. If the information exchange requests event report enabling and the Agent station is prepared to comply, the exchange results in the inclusion of appropriate entries in the Agent's Notification Type table and Event Report Destination table. The Agent responds, indicating to what extent it was able to comply with the enable request. If the exchange requests event report disabling, any corresponding entries in the Agent's tables are removed. Figure 11-3 illustrates this situation for enabling; figure 11-4 for disabling.

Event forwarding disable exchanges can be initiated by an Agent, to indicate that event reports originating from particular managed objects will no longer be sent. The information provided can be directed either to an individual Manager station or to a group of Manager stations for which event forwarding entries exist for those managed objects, and the information exchanged can detail the managed object classes and/or instances whose notifications are no longer to be sent as event reports. The exchange is accompanied by the removal of corresponding entries in the Agent's Notification Type and Event Report Destination tables. Figure 11-5 illustrates this situation.

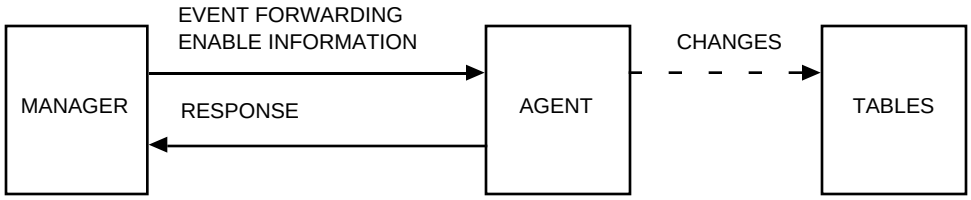


Figure 11-3—Manager-initiated event forwarding enable

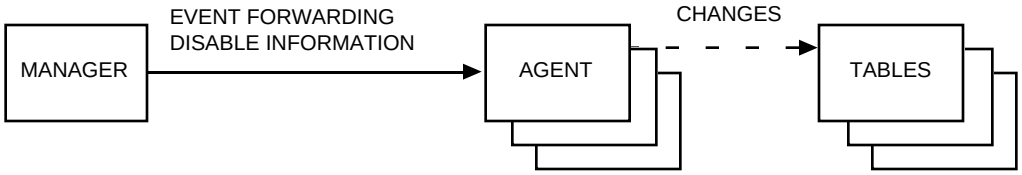


Figure 11-4—Manager-initiated event forwarding disable

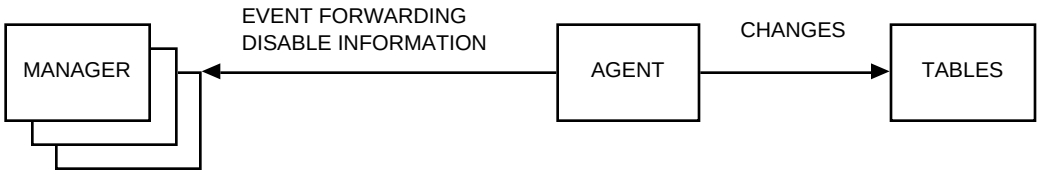


Figure 11-5—Agent-initiated event forwarding disable

11.2.3 Relationship with LMMS/LMMP

The Discovery and Event Forwarding Enable/Disable (DEFED) mechanisms comprise the following components:

a) *DEFED Services.* These services provide additional functionality for LAN/MAN Management over and above the functionality of the LMMS.
b) *DEFED Protocol.* This is a peer protocol, which is exchanged between Manager(s) and Agent(s), and which provides the DEFED services; it makes direct use of the Operation and Notification services of the LMMS, which are in turn provided by the LMMP. The DEFED protocol is therefore exchanged between peer LMMS users.
c) *Agent processing.* The exchange of information involved in DEFED is the result of, or results in, processing that takes place in Agent systems and which manipulates the contents of the Notification Type tables and Event Report Destination tables in those systems. This constitutes an extension to the basic event forwarding mechanisms described in clause 9.
d) *Managed object class definitions and associated management information definitions.* The DEFED protocol is described in terms of management operations and notifications that are part of the definition of a DEFED managed object class. This managed object class represents an extension to the set of LAN/MAN Management managed objects described in clause 8; in particular, it defines additional mechanisms for the manipulation of the table entry managed object classes.

The DEFED mechanisms provide the Manager with a simple means of establishing event report forwarding information in Agent stations, and does not provide for control of the full range of table configurations that is possible using the Notification Type Table Entry and Event Report Destination Table Entry managed object classes defined in clause 8; for example, DEFED deals only with enabling/disabling that relates to managed object instances that an Agent station currently supports, whereas control of the table entry managed objects by means of Get, Set, Create, and Delete operations provides the possibility of establishing forwarding information on the basis of managed object classes for which no instances currently exist in the Agent. Similarly, DEFED cannot control which notification types will be the subject of event reports. DEFED can be used on its own in circumstances where its simple, but limited, control of event forwarding is sufficient, or it can be used in combination with support for the table managed object classes where it is desirable to use the DEFED facilities without losing the greater degree of control that can be achieved by direct manipulation of the table entries.

11.3 Service definition

This subclause defines the services associated with DEFED, following the descriptive conventions defined in ISO/TR 8509 : 1987. The definition of each service includes a table that lists the parameters of its primitives. For a given primitive, the presence of each parameter is described by one of the following values:

M	The parameter is mandatory.
(=)	The value of the parameter is equal to the value of the parameter in the column to the left.
U	The use of the parameter is a service-user option.
—	The parameter is not present in the interaction described by the primitive concerned.
C	The parameter is conditional. The condition(s) are defined by the text that describes the parameter.

11.3.1 Discovery services

There are two Discovery services:

a) Agent Present, which announces the presence of an Agent station, and optionally the managed object classes and instances that it contains.
b) Manager Present, which announces the presence of a Manager station, and optionally the managed object classes and instances in which it is interested.

11.3.1.1 Agent Present service

The Agent Present service is invoked by an Agent station, to announce its presence in the LAN/MAN environment. It allows the Agent to inform potential Managers of its capabilities, in terms of managed object instances that it supports at a given moment in time, and allows the Agent to indicate whether the announcement is unsolicited or is being sent in response to a Manager Present exchange. This service may be used under the following conditions:

a) At power-up or initialization time, to announce the (re)appearance of the station and to indicate its capabilities.
b) When new managed objects are instantiated in the station, to announce the extension of the station’s capabilities and corresponding requirements for event reporting.
c) At any time, to indicate that the Agent station has no established relationship with any Manager station with respect to given managed object instances.
d) In response to the Manager Present service, to provide information solicited by a Manager.

If the Agent Present service is solicited, the Agent Present information exchange takes place between an Agent and the Manager that issued the Manager Present request. In all other cases, the information exchange takes place between an Agent and either an individual Manager or a group of Managers.

The Agent Present service is unconfirmed.

The parameters associated with the Agent Present service are shown in table 11-1.

Table 11-1—Parameters associated with the Agent Present service

Parameter Name	Req/Ind
Solicited	M
Class-Instance List	M

a) **Solicited:** Indicates whether or not the Agent that issues the Agent Present request is doing so of its own volition or as a response to a Manager Present indication primitive. The Solicited parameter can take the values True or False. If the value is True, the Agent Present request was solicited; if the value is False, the request was unsolicited.
b) **Class-Instance List:** Identifies the current capabilities of the Agent that it wishes to announce. The parameter contains an unordered list of managed object classes, and for each managed object class it may optionally include an unordered list of instances of that class. A managed object class shall not appear in the Class-Instance List unless the Agent station currently supports at least one instance of that class. An instance name shall not appear in the Class-Instance List unless it is the name of an instance currently supported by the Agent station. If this parameter contains an empty Class-Instance List, the Agent Present request indicates the presence of an Agent station that currently supports at least one instance of any managed object class.

   NOTE—The ability of an Agent to provide protocol support for the Agent Present request implies at least the presence of an instance of the DEFED managed object class, as defined in 11.5.1.

The level of detail provided by the Agent in the Class-Instance List in an unsolicited request is a local matter, and can provide Station only, Class, or Class and Instance information; however, where the service is being used to provide a solicited response, the level of detail provided shall at least match the level of detail contained in the Class-Instance List of the Manager Present indication that solicited the response.

11.3.1.2 Manager Present service

The Manager Present service is invoked by a Manager station, to announce its presence in the LAN/MAN environment and to inform potential Agents of its capabilities, in terms of managed object classes and/or instances that it is capable of managing; and to solicit responses, in the form of Agent Present service primitives, from Agent stations that will allow it to determine which Agent stations it wishes to manage. The service allows the Manager to indicate whether it seeks responses from all Agent stations, from those with which it has no existing management relationship, or from those with which a management relationship already exists. This service may be used

a) At power-up or initialization time, to announce the (re)appearance of the station and to indicate its capabilities.
b) At any time, to solicit responses from Agent stations with respect to support for particular managed object instances.
c) At any time, to allow the Manager station to confirm which managed object instances in which Agent stations are configured to send it event reports.

Manager Present information exchanges take place between a Manager and either an individual Agent or a group of Agents.

The Manager Present service is unconfirmed.

The parameters associated with the Manager Present service are shown in table 11-2.

Table 11-2—Parameters associated with the Manager Present service

Parameter Name	Req/Ind
Respond If	M
Class-Instance List	M

a) **Respond If:** Indicates whether or not the Agent station(s) that receive the Manager Present indication should respond with respect to the managed object instances indicated by the Class-Instance List parameter. The Respond If parameter can take the following values:
   1) **Event Reporting Enabled**. This value requests the Agent to respond with an Agent Present request if event reporting to the requesting Manager is enabled for any of the managed object instances indicated by the Class-Instance List.
   2) **Event Reporting Disabled**. This value requests the Agent to respond with an Agent Present request if event reporting to the requesting Manager is disabled for any of the managed object instances indicated by the Class-Instance List.
   3) **Event Reporting Enabled Or Disabled**. This value requests the Agent to respond with an Agent Present request for any of the managed object instances indicated by the Class-Instance List, regardless of whether event reporting to the requesting Manager is enabled or disabled.

b) **Class-Instance List:** Identifies the capabilities of the Agent that are of interest to the Manager, in the context established by the Respond If parameter. The parameter contains an unordered list of managed object classes, and for each managed object class it may optionally include an unordered list of instances of that class. The Respond If parameter applies to the set of managed object instances that the Agent currently supports, and which are either explicitly identified by an instance name in the Class-Instance List, or are instances of a class that appears in the list without an accompanying instance list. If the Class-Instance List is empty, the Respond If parameter applies to all the Agent station's currently supported managed object instances.

11.3.2 Event Forwarding Enable/Disable services

There are two Event Forwarding Enable/Disable services:

a) Event Forwarding Enable, which allows a Manager station to request an Agent station to forward event reports to it, and to specify timeout and retry values to be used by the Agent's CPE.
b) Event Forwarding Disable, which allows either a Manager station to request, or an Agent station to signal, termination of forwarding of event reports.

11.3.2.1 Event Forwarding Enable service

The Event Forwarding Enable service is invoked by a Manager station, to request an individual Agent station to insert appropriate entries into its Notification Type table and Event Report Destination table that will permit event reports resulting from notifications emitted by specified managed object instances to be forwarded to the Manager. The Notification Types attribute of any entries created in the Notification Type table

contains an empty list, indicating that the entry applies to all supported notification types. This service may be used whenever the Manager's requirements for event reporting from a given Agent station changes; however, it is typically used following reception of Agent Present indications from an Agent, indicating that a new Agent station has appeared or an existing Agent station has instantiated new managed objects.

The Event Forwarding Enable service is confirmed.

The parameters associated with the Event Forwarding Enable service are shown in table 11-3.

Table 11-3—Parameters associated with the Event Forwarding Enable service

Parameter Name	Req/Ind	Rsp/Conf
CPE Retry Timeout	C	—
CPE Retry Counter	C	—
Class-Instance List	M	—
Character Set	—	M
Enable Response	—	M

a) **CPE Retry Timeout, CPE Retry Counter:** Identifies the retry timeout value, in seconds, and the retry counter value that the Manager wishes the Agent to use in all CPE exchanges with the Manager that use enhanced reliability QOS. These parameters are either both present or both absent as a user option; the absence of these parameters indicates that the Manager wishes the Agent to use basic reliability QOS when issuing event reports to the Manager that originate from the managed object instances identified by the Class-Instance List parameter. The presence of these parameters signals that the Manager wishes the Agent to use enhanced reliability QOS for these event reports.

b) **Class-Instance List:** Identifies the managed object instances currently supported by the Agent from which the Manager wishes to receive event reports. The parameter contains an unordered list of managed object classes, and for each managed object class it may optionally include an unordered list of instances of that class. The Event Forwarding Enable request applies to the set of managed object instances identified by the list. If the list is empty, the request applies to all the Agent station's managed object instances. If the list contains any managed object classes that have no additional instance lists, the request applies to all instances of those classes that the Agent currently supports. If the list contains managed object classes with accompanying instance lists, the request applies to the subset of those instances that the Agent currently supports.

c) **Character Set:** An integer value that identifies the registration number in the International Register of Coded Character Sets for the character set that the Agent supports.

d) **Enable Response:** Indicates, for all elements in the Class-Instance list parameter of the request, which have been successfully enabled for event reporting to the Manager and which have not been enabled, giving the same level of detail as was provided in the Class-Instance List parameter of the request. The parameter takes one of two forms:

   1) **A system enable response code**. This indicates the overall success or failure of the request, and is used in the case where the Class-Instance List parameter is an empty list. It can take one of the following values:

      — **Enabled:** The set of managed object instances supported by the Agent was successfully enabled for event reporting to the Manager.

— **Re-enabled:** The set of managed object instances supported by the Agent was already enabled for event reporting to the Manager.
— **Access Denied:** The request has been rejected because of access restrictions.
— **Resource Limitation:** The request has been rejected because resource limitations in the Agent prevent the insertion of further event report forwarding information in its tables.

2) A **class-instance status list**. This consists of an unordered list of managed object classes, and attached to each managed object class is either a **class enable response code** or a list of managed object instances, where each instance name is accompanied by an **instance enable response code**. The class enable response code form is used to indicate the action taken for all managed object classes in the Class-Instance List parameter of the request for which no instances were specified. The instance enable response code form is used to indicate the action taken for all managed object instances that were explicitly identified in the Class-Instance List parameter of the request.

The **class enable response code** can take the following values:

— **Enabled:** The Agent has enabled event reporting to the Manager for all instances of the specified managed object class that it currently supports.
— **Re-enabled:** The Agent had previously enabled event reporting to the Manager for all instances of the specified managed object class that it currently supports.
— **Class Not Supported:** The Agent currently does not support any instances of the managed object class specified.
— **Access Denied:** The request has been rejected for all instances of the specified managed object class because of access restrictions.
— **Resource Limitation:** This part of the request has been rejected because resource limitations in the Agent prevent the insertion of further event report forwarding information in its tables.

The **instance enable response code** can take the following values:

— **Enabled:** The Agent has enabled event reporting to the Manager for the managed object instance specified.
— **Re-enabled:** The Agent had previously enabled event reporting to the Manager for the managed object instance specified.
— **Instance Not Supported:** The managed object instance specified is not currently supported by the Agent.
— **Access Denied:** The request has been rejected for this managed object instance because of access restrictions.
— **Resource Limitation:** This part of the request has been rejected because resource limitations in the Agent prevent the insertion of further event report forwarding information in its tables.

11.3.2.2 Event Forwarding Disable service

The Event Forwarding Disable service is invoked

a) By a Manager station, to request an individual Agent or a group of Agents to disable event reporting to that Manager with respect to specified managed object instances. The Manager may use this service at any time.
b) By an Agent station, to inform an individual Manager or a group of Managers that event reporting has been disabled with respect to specified managed object instances. The Agent may use this service at any time, but typically will use this service when the deletion of managed objects renders obsolete some of the information in its Notification Type table and Event Report Destination table;

under these circumstances, the Agent is responsible for removing the redundant information from the tables.

The Event Forwarding Disable service is unconfirmed.

The parameters associated with the Event Forwarding Disable service are shown in table 11-4.

Table 11-4—Parameters associated with the Event Forwarding Disable service

Parameter Name	Req/Ind
Originator	M
Class-Instance List	M

a) **Originator:** Identifies whether the originator of the Event Forwarding Disable request was a Manager or an Agent. The parameter can take two values: **Manager** or **Agent**.
b) **Class-Instance List:** Identifies the managed object instances in the Agent that are the subject of the Event Forwarding Disable request. The parameter contains an unordered list of managed object classes, and for each managed object class it may optionally include an unordered list of instances of that class. The Event Forwarding Disable request applies to the set of managed object instances identified by the list. If the Class-Instance List is empty, the request applies to all the managed objects that the Agent station currently supports. If the list contains any managed object classes that have no additional instance lists, the request applies to all instances of those classes that the Agent currently supports. If the list contains managed object classes with accompanying instance lists, the request applies to those instances. The meaning of the Class-Instance List depends upon the value of the Originator parameter, as follows:
    1) If the Originator is a Manager, the service constitutes a request from the Manager station to one or more Agent stations to remove event forwarding information from their Notification Type tables and Event Report Destination tables so as to disable event reporting to that Manager with respect to the set of managed object instances identified by the Class-Instance List.
    2) If the Originator is an Agent, the service constitutes an announcement from the Agent station to one or more Manager stations that its Notification Type table and Event Report Destination table have been modified so as to disable event reporting with respect to the set of managed object instances identified by the Class-Instance List.

11.3.3 Use of the service primitives

This subclause gives some examples of the use of the Discovery and Event Forwarding Enable/Disable service primitives in different situations, and shows the time sequence in which the primitives are invoked in each case. The examples are not exhaustive, nor are they statements of mandatory behavior; they are intended to illustrate how the services might be used in given scenarios.

In the time sequence diagrams shown, time is assumed to progress along the vertical axis of the page; events that are nearest the top of the page occur before events that appear further down the page.

11.3.3.1 Agent initialization

This example shows the use of the services in the situation where an Agent station is initialized, for example as a result of power-up or reboot, and has no preconfigured information that determines which Manager(s) it should send event reports to. The sequence of events is as follows:

a) The Agent invokes the Agent Present service to announce its presence to all Managers, using the Unsolicited variant of the service with an empty Class-Instance List parameter.
b) The level of detail provided in the Agent Present indication (in terms of managed object instances supported by the Agent) is insufficient for the Manager(s) to determine whether or not to receive the Agent's event reports, so the Manager(s) use the Manager Present service to solicit more detail, and the Agent responds using the Solicited variant of the Agent Present service, providing details of supported managed object instances. (These exchanges would be omitted if the detail provided in the original Agent Present exchange was sufficient for the Manager(s) to reach a decision.)
c) The Manager(s) that wish to receive event reports from the Agent use the Event Forwarding Enable service to request the Agent to set up the appropriate table information to allow the event reports to be sent to the manager(s) for those managed object instances that are of interest.

Figure 11-6 shows the sequence of service primitives involved, assuming two Managers, one of which requests event report enabling.

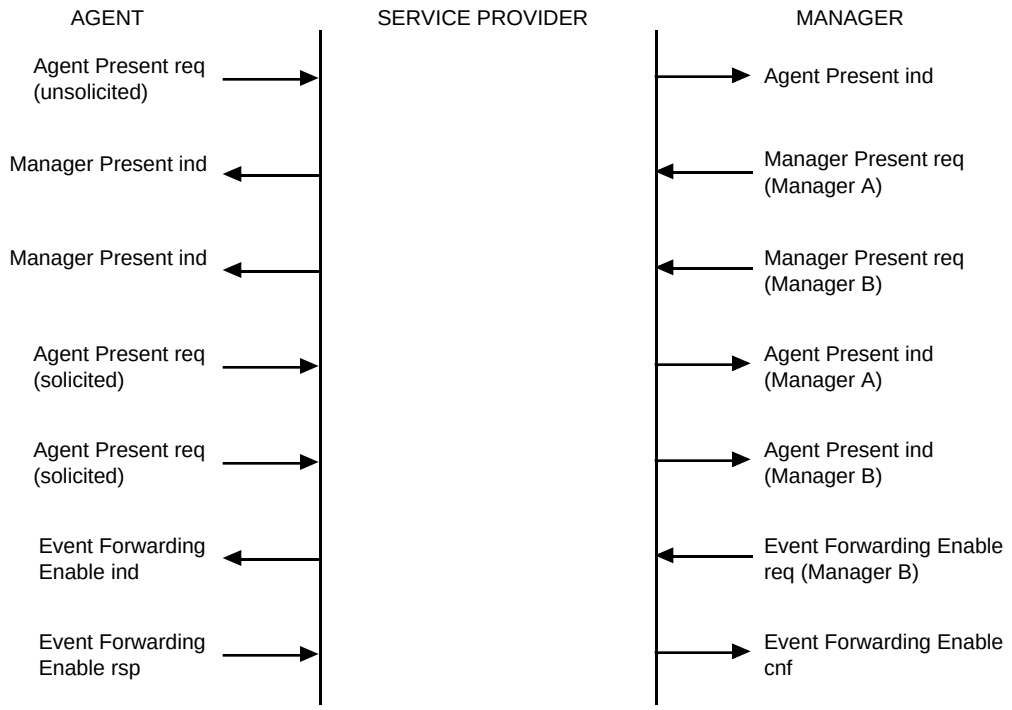


Figure 11-6—Agent initialization time sequence

11.3.3.2 Manager initialization

This example shows the use of the services in the situation where a Manager station is initialized, for example as a result of power-up or reboot, and cannot assume that the Agent(s) have preconfigured information that determines which Manager(s) they should send event reports to. The sequence of events is as follows:

a) The Manager uses the Manager Present service to solicit responses from Agent systems, and provides a Class-Instance list that details the managed object instances that are of interest to it.
b) The Agent(s) respond using the Solicited variant of the Agent Present service, providing details to the Manager of supported managed object instances that constitute a subset of those specified by the Manager.

c) The Manager uses the Event Forwarding Enable service to request the Agent(s) that it wishes to receive event reports from to set up the appropriate table information to allow the event reports to be sent to the manager(s) for those managed object instances that are of interest.

Figure 11-7 shows the sequence of service primitives involved, assuming two Agents, one of which does not support any managed object classes/instances that the Manager is interested in.

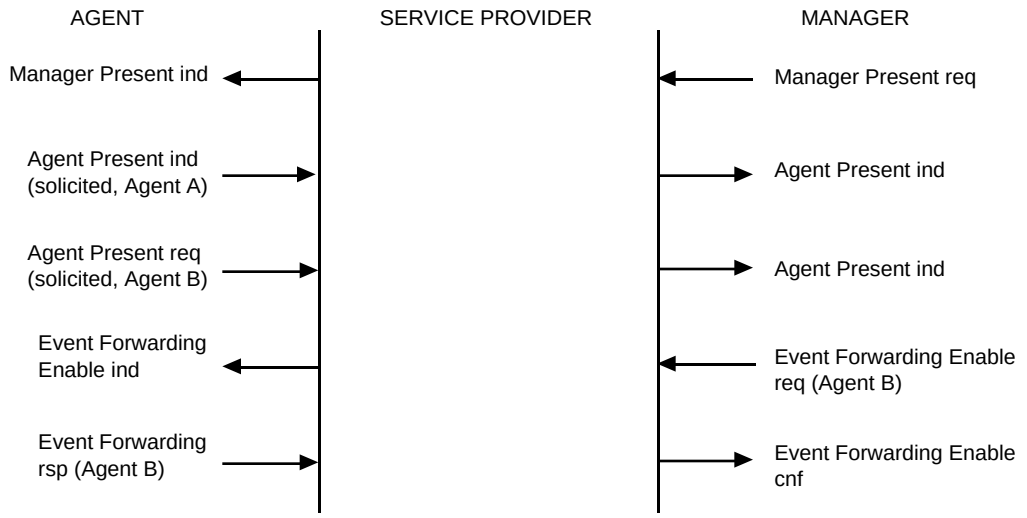


Figure 11-7—Manager initialization time sequence

11.3.3.3 Agent configuration change

This example shows the use of the services in the situation where an Agent station's configuration changes in such a way that some of its managed objects have no Manager configured to receive event reports; for example, as a result of managed objects being created, or a Manager using the Event Forwarding Disable service. The sequence of events is as follows:

a) The Agent invokes the Agent Present service to announce its presence to all Managers, using the Unsolicited variant of the service with a Class-Instance List parameter that carries the relevant managed object instance information.
b) The Manager(s) that wish to receive event reports from the Agent use the Event Forwarding Enable service to request the Agent to set up the appropriate table information to allow the event reports to be sent to the manager(s) for those managed object instances that are of interest.

Figure 11-8 shows the sequence of service primitives involved, assuming two Managers, both of which request enabling of event forwarding.

11.3.3.4 Manager configuration check

This example shows the use of the services in the situation where a Manager wishes to check that its own picture of the network matches with current reality; for example, to check that all Agent stations from which it is expecting to receive event reports are still active. The sequence of events is as follows:

a) The Manager invokes the Manager Present service to solicit responses from all Agents for which event report forwarding is enabled to it.

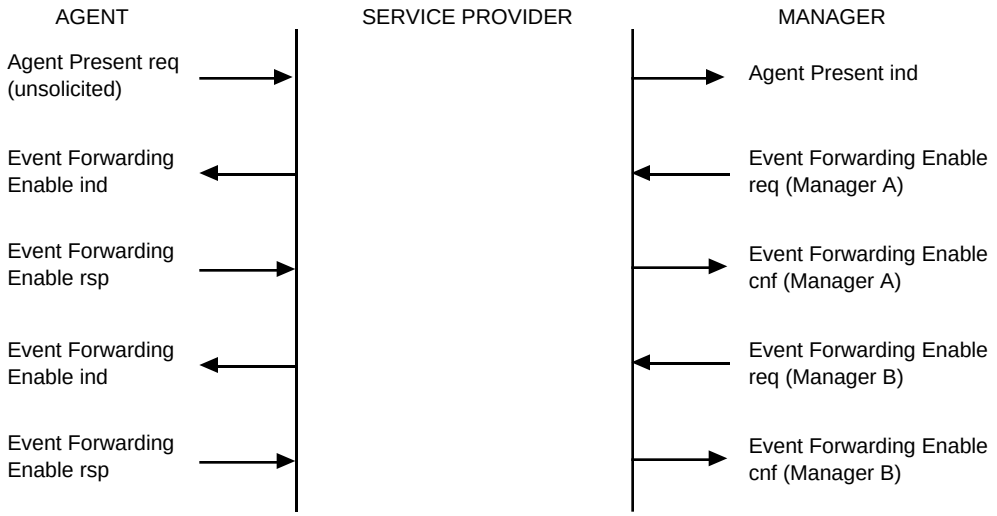


Figure 11-8—Agent configuration change time sequence

b) The Agent(s) invoke the Agent Present service to return the solicited information to that Manager, using the Solicited variant of the service with a Class-Instance List parameter that carries the relevant instance information.

Figure 11-9 shows the sequence of service primitives involved, assuming two Agents, both of which respond.

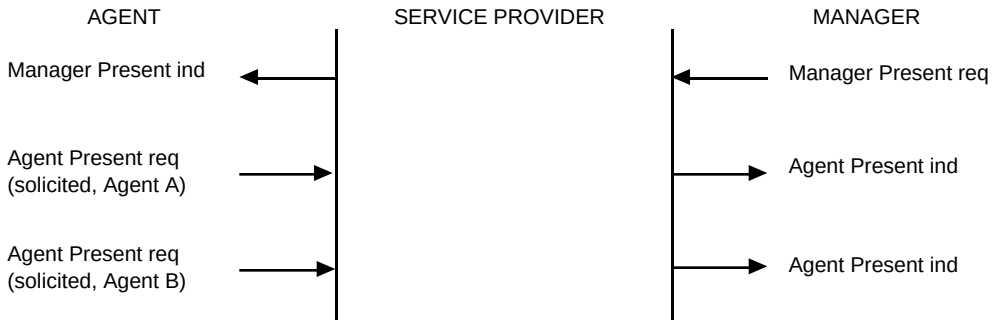


Figure 11-9—Manager configuration check time sequence

11.3.3.5 Disabling event forwarding

This example shows the use of the services in the situation where a Manager wishes to disable event forwarding for particular managed object instances, or an Agent wishes to report to a Manager that it will no longer be forwarding event reports related to particular managed object instances. In both cases, the Manager or Agent station makes use of the Event Forwarding Disable service to achieve the desired result.

If the use of the Event Forwarding Disable service by a Manager results in one or more managed objects that are no longer enabled for event reporting to any manager, the Agent may then issue an Agent Present event report to initiate the sequence described in 11.3.3.3, in order to rectify this situation.

Figures 11-10 and 11-11 show the sequence of service primitives involved in disabling event forwarding.

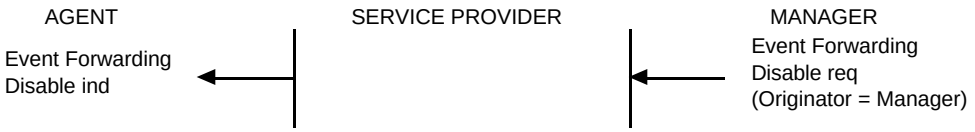


Figure 11-10—Manager disable time sequence

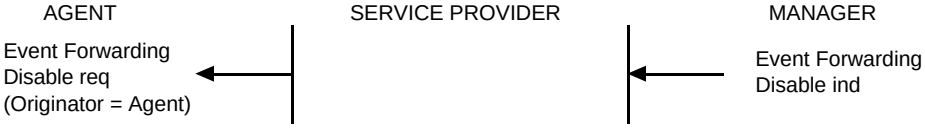


Figure 11-11—Agent disable time sequence

NOTE—Agent stations make use of the Event Forwarding Disable service only when disabling event report forwarding as a result of their own decisions, and do not use the service to confirm that event reporting has been disabled following receipt of an Event Forwarding Disable indication from a Manager.

11.3.3.6 Use of DEFED and LMMS in combination

Where an Agent station supports both the DEFED mechanisms and the Notification Type Table Entry and Event Report Destination Table Entry managed object classes (defined in 8.6, 8.7, and 11.6) that provide management access to the tables, the DEFED services can be used in combination with the other services available from the LMMS in order to provide more sophisticated control of event reporting. The LMMS services that are relevant are the following:

a) M-CREATE, which can be used in order to create entries in the tables that have particular characteristics. For example, this service could be used in place of the Event Forwarding Enable service in the examples above, to create a Notification Type Table Entry managed object that would permit the forwarding of particular notification types from all instances (present and future) of a given managed object class.
b) M-DELETE and M-SET, which can be used in order to modify or remove information from the tables. For example, this could be used in place of the Event Forwarding Disable service in the examples above, to permit one Manager to remove table entries that apply to any other Manager (assuming that its access rights permitted the operation);
c) M-GET, which can be used in order to examine the contents of the tables. For example, a Manager might use M-GET in place of the Manager Present service in order to examine forwarding information that applies to other Managers.

11.3.3.7 Iterative enabling of event forwarding

This example shows the use of the services in the situation where a Manager station wishes to establish event reporting relationships with a large number of Agent stations. In such circumstances, the Agent Present event reports received by the Manager may be too numerous for it to handle at one attempt; the following demonstrates how the DEFED services may be used in an iterative manner. The sequence of events is as follows:

a) The Manager uses the Manager Present service to solicit responses from all Agent systems, and provides a Class-Instance list that details the managed object instances that are of interest to it. The Respond If parameter indicates that responses are solicited from Agents that are not enabled for those managed object instances.

b) The Agent(s) respond using the Solicited variant of the Agent Present service, providing details to the Manager of supported managed object instances that constitute a subset of those specified by the Manager.
c) The Manager uses the Event Forwarding Enable service to enable event forwarding for those instances from as many Agents as it was able to receive responses from.
d) The Manager repeats the above process until no Agent Present indications are received following the Manager Present request.

Figure 11-12 shows the sequence of service primitives involved, assuming 26 Agent stations (A through Z), and the Manager successfully receives Agent Present indications from A through Q on the first iteration, catching the remainder on the second iteration.

11.3.3.8 Service limitations

It should be noted that the DEFED services are provided by means of the LMMS and its underlying Convergence service; the provision of the LMMS by the LMMPE and CPE places restrictions on the LMM-PDU size that can be accommodated, as described in 7.3.6.2. The consequence of this is that users of the DEFED services that wish to exchange larger volumes of information than can be accommodated in a single service request must split that information across a number of service requests. For example, an Agent station wishing to announce its presence and to include all of its current instance information would issue as many Agent Present service requests as is necessary in order to accommodate the instance information, each request giving a subset of the list of instances that the Agent supports.

11.4 Protocol specification

This subclause describes how the Discovery and Event Forwarding Enable/Disable services are provided by the DEFED protocol entity.

11.4.1 Provision of the Agent Present service

The Agent Present service is used by the Agent when the DEFED managed object issues an AgentPresent notification that is required to be delivered to one or more Managers as an Agent Present event report. The service is provided by use of the LMMS M-EVENT-REPORT service, as follows.

a) On receipt of an Agent Present request primitive, the DEFED protocol entity issues an M-EVENT-REPORT request, with the following parameter values:
    1) **Invoke identifier:** Assigned a value that uniquely identifies this Event Report.
    2) **Mode:** Assigned the value nonconfirmed.
    3) **Managed object class:** Assigned the class name of the DEFED managed object class, as defined in 11.5.
    4) **Managed object instance:** Assigned the instance name of the DEFED managed object.
    5) **Event type:** Assigned the value of the AgentPresent notification type, as defined in 11.5.6.
    6) **Event time:** May be used, as a user option.
    7) **Event information:** Assigned a value derived from the Solicited and Class-Instance List parameters of the Agent Present request primitive. The abstract syntax of this value is described by the ASN.1 data type AgentPresent defined in 11.7. The Solicited variant of this data type shall be used if the value of the Solicited parameter is equal to True; the Unsolicited variant shall be used if the value of the Solicited parameter is equal to False. The data type ObjectClassInstance is used to represent the Class-Instance List parameter value. If the parameter contains an empty list, this is represented in the AgentPresent data type as an empty SET OF ObjectClassInstance.

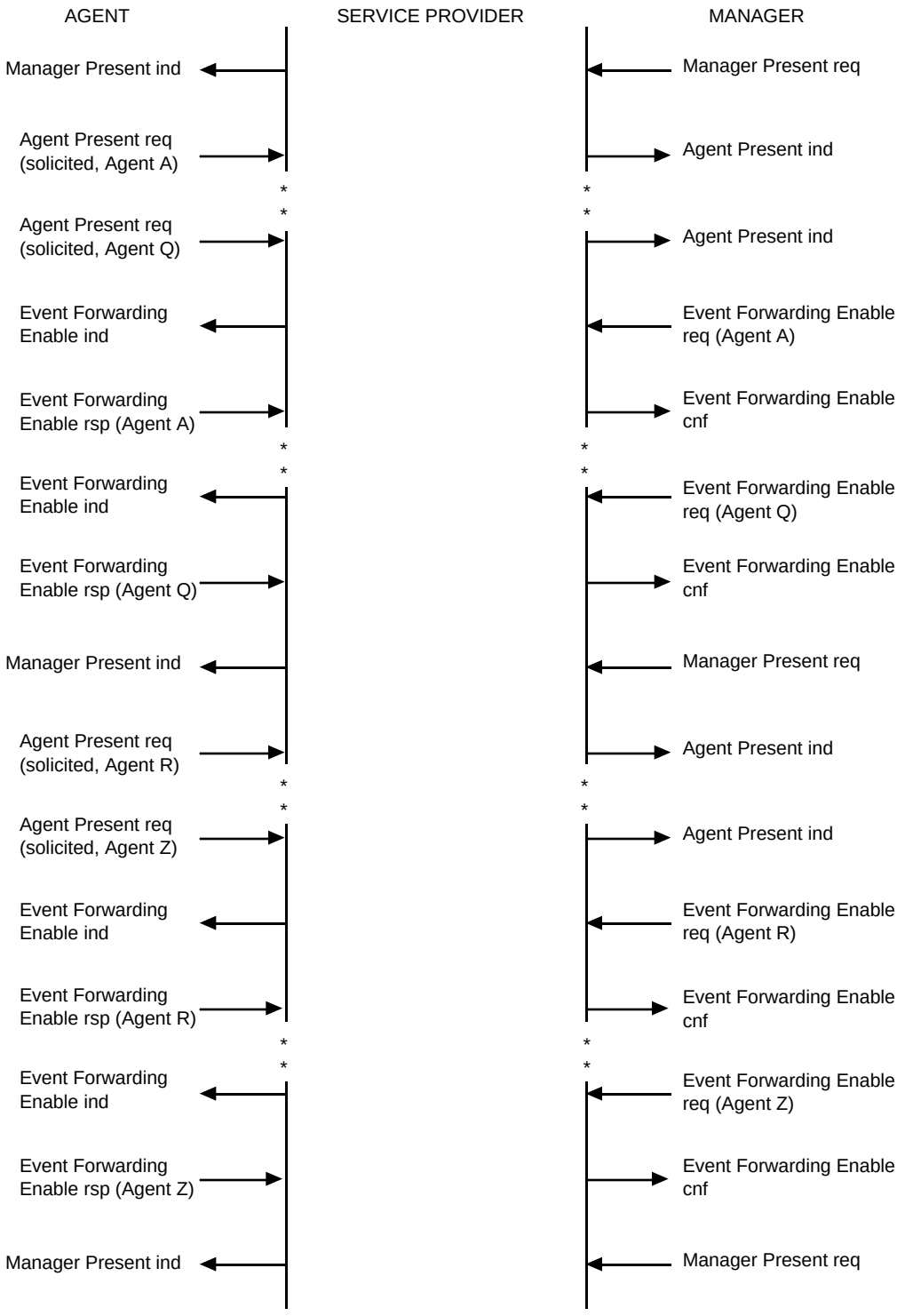


Figure 11-12—Iterative Event Forwarding Enable time sequence

b) On receipt of an M-EVENT-REPORT indication in which the Event type parameter is equal to the value of the AgentPresent notification type, the DEFED protocol entity issues an Agent Present indication primitive to the Manager, with parameters derived from the Event Information parameter as described above.

11.4.2 Provision of the Manager Present service

The Manager Present service is provided by use of the LMMS M-ACTION service, as follows.

a) On receipt of a Manager Present request primitive, the DEFED protocol entity issues an M-ACTION request, with the following parameter values:
   1) **Invoke identifier:** Assigned a value that uniquely identifies the Action.
   2) **Mode:** Assigned the value nonconfirmed.
   3) **Base object class:** Assigned the class name of the DEFED managed object class, as defined in 11.5.
   4) **Base object instance:** Assigned the instance name of the DEFED managed object.
   5) **Scope, Filter:** Not used.
   6) **Access control:** Used at the Manager's option.
   7) **Synchronization:** Not used.
   8) **Action type:** Assigned the value of the ManagerPresent action type, as defined in 11.5.5.
   9) **Action information:** Assigned a value derived from the Respond If and Class-Instance List parameters of the Manager Present request primitive. The abstract syntax of this value is described by the ASN.1 data type ManagerPresent defined in 11.7. The value of the Respond If parameter determines the value of the RespondIf data type. The data type ObjectClassInstance is used to represent the Class-Instance List parameter value. If the parameter contains an empty list, this is represented in the AgentPresent data type as an empty SET OF ObjectClassInstance.

b) On receipt of an M-ACTION indication in which the Action type parameter is equal to the value of the ManagerPresent action type, the DEFED protocol entity issues a Manager Present indication primitive to the Agent, with parameters derived from the Action Information parameter as described above. This causes the Agent to send a Manager Present action operation to the DEFED managed object.

11.4.3 Provision of the Event Forwarding Enable service

The Event Forwarding Enable service is provided by use of the LMMS M-ACTION service, as follows.

a) On receipt of an Event Forwarding Enable request primitive, the DEFED protocol entity issues an M-ACTION request, with the following parameter values:
   1) **Invoke identifier:** Assigned a value that uniquely identifies the Action.
   2) **Mode:** Assigned the value confirmed.
   3) **Base object class:** Assigned the class name of the DEFED managed object class, as defined in 11.5.
   4) **Base object instance:** Assigned the instance name of the DEFED managed object.
   5) **Scope, Filter:** Not used.
   6) **Access control:** Used at the Manager's option.
   7) **Synchronization:** Not used.
   8) **Action type:** Assigned the value of the EFEnable action type, as defined in 11.5.3.
   9) **Action information:** Assigned a value derived from the CPE Retry Timeout, CPE Retry Counter, and Class-Instance List parameters of the Event Forwarding Enable request primitive. The abstract syntax of this value is described by the ASN.1 data type EFEnable defined in 11.7. The value of the CPE Retry Timeout parameter determines the value of the cpeRetryTimeout field. The value of the CPE Retry Counter parameter determines the value of the cpeRetryCounter field. If the timeout and counter parameters are absent, both fields are absent.

The classInstance field is used to represent the Class-Instance List parameter value. If the parameter contains an empty list, this is represented as an empty SET OF ObjectClassInstance.

b) On receipt of an M-ACTION indication in which the Action type parameter is equal to the value of the EFEnable action type, the DEFED protocol entity issues an Event Forwarding Enable indication primitive to the Agent, with parameters derived from the Action information parameter as described above. This causes the Agent to send an Event Forwarding Enable action operation to the DEFED managed object.

c) On completion of the Event Forwarding Enable action operation by the DEFED managed object, the Agent issues an Event Forwarding Enable response primitive, whose parameter values are determined by the result of the operation. This causes the DEFED protocol entity to issue one or more M-ACTION responses. If necessary in order to comply with the PDU size limitations of the protocol, the DEFED protocol entity is permitted to issue multiple responses to the action operation, each one of which represents a consistent subset of the results of the operation. The requirement for consistency in multiple replies is that any one such response shall be consistent with the response that would have been generated if the incoming request had requested enabling of the specific set of managed object classes/instances that are identified in that one response.

In the case where a single response is generated, the parameter values used in the response are as follows:

1) **Invoke identifier:** This parameter is assigned the value of the corresponding parameter in the M-ACTION indication primitive.
2) **Linked identifier:** Not present.
3) **Managed object class, Managed object instance, Action type, Current time:** May be used, as a user option.
4) **Action reply:** If the action was not in error, this parameter is assigned a value derived from the Character Set and Enable Response parameters of the Event Forwarding Enable response primitive. The abstract syntax of this value is described by the ASN.1 data type EFEnableResponse defined in 11.7. The value of the Character Set parameter determines the value of the characterSet field. The classOrCode field is used to represent the Enable Response parameter value.
5) **Errors:** Used if the received action was in error.

In the case where multiple responses are generated, the parameter values used in the responses are as follows:

1) **Invoke identifier:** In the final response, this parameter is assigned the value of the corresponding parameter in the M-ACTION indication primitive. In all other responses, distinct values are used that uniquely identify each response.
2) **Linked identifier:** Not present in the final response. In all other responses, this parameter is assigned the value of the Invoke Identifier parameter in the M-ACTION indication primitive.
3) **Managed object class, Managed object instance, Action type, Current time:** May be used, as a user option.
4) **Action reply:** In the final response, this parameter is not used. In all other responses, this parameter is assigned a value derived from the Character Set and Enable Response parameters of the Event Forwarding Enable response primitive. The abstract syntax of this value is described by the ASN.1 data type EFEnableResponse defined in 11.7. The value of the Character Set parameter determines the value of the characterSet field. The classOrCode field is used to represent a subset of the Enable Response parameter value, providing the results that relate to one or more of the managed object classes/instances that are the subject of the response.
5) **Errors:** Not used.

d) On receipt of an M-ACTION confirmation in which the Action Type parameter is equal to the value of the EFEnable action type, the DEFED protocol entity issues an Event Report Enable confirmation

to the Manager, with Character Set and Enable Response parameters derived from the Action reply parameter(s) of the M-ACTION confirmation(s), as described above.

11.4.4 Provision of the Event Forwarding Disable service

The service is used by the Agent when the DEFED managed object issues an EFDisable notification that is required to be delivered to one or more Managers as an EFDisable event report, or by a Manager when it wishes to send an EFDisable action to one or more Agents. The Event Forwarding Disable service is provided by use of the LMMS M-EVENT-REPORT and M-ACTION services, as follows.

a) On receipt of an Agent Present request primitive with Originator parameter equal to "Agent," the DEFED protocol entity issues an M-EVENT-REPORT request, with the following parameter values:
   1) **Invoke identifier:** Assigned a value that uniquely identifies this Event Report.
   2) **Mode:** Assigned the value nonconfirmed.
   3) **Managed object class:** Assigned the class name of the DEFED managed object class, as defined in 11.5.
   4) **Managed object instance:** Assigned the instance name of the DEFED managed object.
   5) **Event type:** Assigned the value of the EFDisable notification type, as defined in 11.5.7.
   6) **Event time:** May be used, as a user option.
   7) **Event information:** Assigned a value derived from the Class-Instance List parameter of the Event Forwarding Disable request primitive. The abstract syntax of this value is described by the ASN.1 data type EFDisable defined in 11.7. The data type ObjectClassInstance is used to represent the Class-Instance List parameter value. If the parameter contains an empty list, this is represented in the AgentPresent data type as an empty SET OF ObjectClassInstance.

b) On receipt of an M-EVENT-REPORT indication in which the Event type parameter is equal to the value of the EFDisable notification type, the DEFED protocol entity issues an Event Forwarding Disable indication primitive to the Manager, with Originator parameter set to the value "Agent" and Class-Instance List parameter derived from the Event Information parameter as described above.

c) On receipt of an Event Forwarding Disable request primitive with Originator parameter equal to "Manager," the DEFED protocol entity issues an M-ACTION request, with the following parameter values:
   1) **Invoke identifier:** Assigned a value that uniquely identifies the Action.
   2) **Mode:** Assigned the value nonconfirmed.
   3) **Base object class:** Assigned the class name of the DEFED managed object class, as defined in 11.5.
   4) **Base object instance:** Assigned the instance name of the DEFED managed object.
   5) **Scope, Filter:** Not used.
   6) **Access control:** Used at the Manager's option.
   7) **Synchronization:** Not used.
   8) **Action type:** Assigned the value of the EFDisable action type, as defined in 11.5.4.
   9) **Action information:** Assigned a value derived from the Class-Instance List parameter of the Event Forwarding Disable request primitive. The abstract syntax of this value is described by the ASN.1 data type EFDisable defined in 11.7. The data type ObjectClassInstance is used to represent the Class-Instance List parameter value. If the parameter contains an empty list, this is represented in the AgentPresent data type as an empty SET OF ObjectClassInstance.

d) On receipt of an M-ACTION indication in which the Action type parameter is equal to the value of the EFDisable action type, the DEFED protocol entity issues an Event Forwarding Disable indication primitive to the Agent, with Originator parameter set to the value "Manager" and Class-Instance List parameter derived from the Event Information parameter as described above. This causes the Agent to send an EFDisable action operation to the DEFED managed object.

11.4.5 Use of individual and group addressing

The destination addresses used in DEFED protocol exchanges may identify an individual destination station or a group of destination stations, as permitted by the service being provided. When initiating a protocol exchange by use of the LMMS, the DEFED protocol entity uses the Destination Address parameter of the CPE-Data service in order to specify the destination address to be used. The address provided is a CPE address in which the MAC address component of the address determines whether the destination will be an individual station or a group of stations; an individual MAC address is used when the destination is an individual station, a group MAC address is used when the destination is a group of stations. Two standardized group MAC addresses are documented in clause 12 for use with LAN/MAN Management; these group MAC addresses can be used as destination addresses in DEFED protocol exchanges where the destination group is all Manager stations or all Agent stations.

11.4.6 Forwarding of Agent Present and Event Forwarding Disable event reports

The Agent behaves as if event forwarding is permanently enabled with respect to Agent Present and Event Forwarding Disable notifications. The consequence of this is that the contents of the entries in the Notification Type Table and Event Report Destination table has no effect upon the ability of the Agent to participate in DEFED protocol exchanges with any Manager or group of Managers.

11.4.7 Table housekeeping

The possibility exists with the DEFED protocol that an Agent station may accumulate obsolete information in its Notification Type table and Event Report Destination table, as a consequence of a Manager being removed from the network. Under these circumstances, there are two mechanisms whereby its tables may be purged of such information:

a) If the Agent supports access to these tables via the table entry managed object classes defined in 8.6, 8.7, and 11.6, the LMMS services could be used by a remote Manager to remove obsolete information.
b) If the Agent issues an event report and makes use of the CPE's enhanced reliability QOS, a resulting timeout can be used to signal to the Agent that the Manager is no longer available. The Agent could then remove the relevant entries from its tables.

The use of either or both of these mechanisms for table housekeeping is an implementation option.

11.5 Management information definitions for DEFED

This subclause contains the definition of the Discovery and Event Forwarding Enable/Disable managed object class and its associated management information definitions, defined using the Template notation described in ISO/IEC 10165-4 : 1992.

11.5.1 DEFED managed object class

This managed object class is required in order to provide management access to the Agent functionality associated with the Discovery and Event Forwarding Enable/Disable mechanisms. It defines the relationships between the DEFED notifications/actions and the Notification Type table and Event Report Destination table. A single instance of the DEFED managed object class may exist within the LAN/MAN Management managed object class (see 8.2).

Support of this managed object class is mandatory for implementations claiming to support the DEFED mechanisms as an Agent.

```
oDEFED MANAGED OBJECT CLASS
   DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
   CHARACTERIZED BY
      pDEFED PACKAGE
         ATTRIBUTES       aDEFName ;
         ACTIONS          acEFEnable,
                          acEFDisable,
                          acManagerPresent;
         NOTIFICATIONS    nAgentPresent,
                          nEFDisable ;
      ;
   ;
REGISTERED AS  {iso(1)member-body(2)us(840)ieee802-1B(10007)managedObjectClass(3)
               defedMO(5)};

nDEFED NAME BINDING
   SUBORDINATE OBJECT   CLASS         oDEFED AND SUBCLASSES;
   NAMED  BY SUPERIOR  OBJECT CLASS   oLANMANManagement AND SUBCLASSES;
   WITH ATTRIBUTE                     aDEFName;
   BEHAVIOUR
      bDEFEDnb BEHAVIOUR
         DEFINED AS    !A single instance of the oDEFED managed object class exists
                       within the oLANMANManagement managed object class.!
      ;
   ;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007) nameBinding(6)
               defednamebinding(6)};
```

11.5.2 DEFName attribute

```
aDEFName ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-DEFDefinitions.DEFName;
   MATCHES FOR EQUALITY;
   BEHAVIOUR
      bDEFName BEHAVIOUR
         DEFINED AS    !This attribute is used to name instances of the Discovery
                       and Event Forwarding Enable/Disable managed object class. The
                       value of this attribute is fixed and is equal to the string
                       "DEF".!
      ;
   ;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7)
               defname(14)};
```

11.5.3 EFEnable action

```
acEFEnable ACTION
   BEHAVIOUR
      bEFEnable BEHAVIOUR
         DEFINED AS    !This action requests that:
                       -     the originating Manager be sent event reports resulting
                         from notifications emitted by a particular set of managed
                         object instances;
                       -     CPE requests destined for the originating Manager that use
                         the enhanced QOS option use particular values of retry timeout
                         and retry counter;
                       -     Event reports originating from the identified set of
                         managed object instances and destined for the originating
                         Manager be delivered using a particular CPE QOS option.

                       If  there is sufficient capacity in the  Notification Type
                       table and Event Report Destination table to  do  so, and if
                       there are no access control restrictions, the effect of this
                       action is to include entries in the table(s) that will permit
                       event reporting to the Manager that originated the action for
                       all notifications emitted by the set of managed object
                       instances currently supported by the Agent that were specified
                       in the action request,and to include the retry timeout and
                       counter values in the Event Report Destination table entry for
```

```
                    that Manager.  The creation of new entries to represent this
                    information in the various tables occurs only if the tables
                    are not resource constrained and the information concerned
                    does not already exist in the tables. Managed objects that
                    correspond to these entries are only created if the Agent
                    system supports the managed object classes concerned.
                    Modifications to the tables are carried out as follows:

                    -    An entry is created in the Event Report Destination
                      table that contains the CPE address of the Manager that
                      originated the action. If the Agent supports enhanced
                      reliability QOS, and the action request included retry and
                      timeout information, the Destination QOS attribute is set
                      to enhanced reliability; otherwise it is set to basic
                      reliability.
                    -    If the Action request specified retry and timeout
                      information, an entry in the Specific CPE Info table is
                      created, with CPE Address set to the CPE address of the
                      Manager that originated the action, and Default CPE Info set
                      to represent the timeout and retry values carried in the
                      Action request.
                    -    The set of managed object instances that are eligible
                      to be enabled, the "eligible set," is determined, by
                      evaluating the set intersection between the set of managed
                      object instances that the Agent currently supports, the
                      "supported set," and the set of managed object instances
                      which the Manager requested to be enabled, the "requested
                      set." The determination of the membership of the requested
                      set is as follows:

                      (1) If the Action request contained an empty Class-Instance
                      List, the requested set is equal to the supported set.

                      (2) If the Action request contained any managed object
                      classes for which no instance names were specified, the
                      requested set includes all members of the supported set
                      whose managed object class matches those specified in the
                      request.

                      (3) If the Action request contained any managed object
                      classes for which instance names were specified, the
                      requested set includes all members of the supported set
                      whose managed object class and instance match those
                      specified in the request. If the class specified in the
                      request was the "actual class" managed object class name
                      defined in ISO/IEC 10165-4 : 1992, then the requested set
                      contains all members of the supported set whose instance
                      names match those that were specified for the "actual class"
                      entry in the request.
                    -    An entry is created in the Notification Type table for
                      each distinct managed object class that is represented by
                      the eligible set, with the class name attribute set to the
                      value of the class name and the managed object instances
                      attribute set to the name(s) of the instances of that class
                      that appear in the eligible set. The Notification Types
                      attribute in each entry contains the empty set, and the
                      Event Report Destinations attribute points to the Event
                      Report Destination table entry that identifies the
                      originating Manager.

                    The results returned by the operation indicate which managed
                    object instances were actually enabled, and what character set
                    is supported by the station. !
      ;
   ;
   MODE CONFIRMED ;
   WITH INFORMATION SYNTAX IEEE802-1-DEFDefinitions.EFEnable ;
   WITH REPLY SYNTAX IEEE802-1-DEFDefinitions.EFEnableResponse;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007) action(9)
               efenable(0)};
```

11.5.4 EFDisable action

```
acEFDisable ACTION
   BEHAVIOUR
      bEFDisable BEHAVIOUR
         DEFINED AS   !This action is sent by a Manager to request that a station
                      modify its Notification Type table and Event Report
                      Destination table to prevent the Agent from forwarding
                      notifications emitted by specified managed object instances
                      to that Manager. If the Agent's tables contain forwarding
                      information that relates to those managed object instances and
                      to the originating Manager, the entries are removed, as
                      follows.

                      -    The set of managed object instances that are eligible
                        to be disabled, the "eligible set," is determined, by
                        evaluating the set intersection between the set of managed
                        object instances that the Agent currently supports, the
                        "supported set," and the set of managed object instances
                        that the Manager requested to be disabled, the "requested
                        set." The determination of the membership of the requested
                        set is as follows:
                        1) If the Action request contained an empty Class-Instance
                        List, the requested set is equal to the supported set.
                        2) If the Action request contained any managed object
                        classes for which no instance names were specified, the
                        requested set includes all members of the supported set
                        whose managed object class matches those specified in the
                        request.
                        3) If the Action request contained any managed object
                        classes for which instance names were specified, the
                        requested set includes all members of the supported set
                        whose managed object class and instance names match those
                        specified in the request. If the class specified in the
                        request was the "actual class" managed object class name
                        defined in ISO/IEC 10165-4 : 1992, then the requested set
                        contains all members of the supported set whose instance
                        names match those that were specified for the "actual class"
                        entry in the request.
                      -    The Notification Type table is searched for all entries
                        which specify managed object instance names that identify
                        members of the eligible set, and which contain index values
                        in the Event Report Destinations field that point to entries
                        in the Event Report Destination table that contain the
                        address of the originating Manager. For each entry found,
                        check the following:
                        1) If the set of managed object instances identified by the
                        table entry is a subset of the eligible set, the originating
                        manager's index value is removed from the Event Report
                        Destinations field. If this results in an empty Event Report
                        Destinations field, the table entry is removed.
                        2) If the table entry identifies managed object instances
                        that are not members of the eligible set, two possibilities
                        exist:
                            a) The entry specifies the originating Manager as the
                            only destination index. In this case, the table entry
                            is modified to remove the instance names that are
                            members of the eligible set.
                            b) The entry specifies two or more different
                            destinations. In this case, the index that relates to
                            the requesting manager is removed. If resources permit,
                            a new Notification Type table entry is created, by
                            taking a copy of the existing table entry, setting the
                            Event Report Destinations field to point only at the
                            originating Manager, and then reapplying the rules
                            described in a).

                      -    If the application of the above rules results in the
                        originating Manager's entry or entries in the Event Report
                        Destination table no longer being referenced from the
                        Notification Type table, the entry or entries are removed. !
```

```
    ;
  ;
  WITH INFORMATION SYNTAX IEEE802-1-DEFDefinitions.EFDisable;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) action(9)
              efdisableAction(1)};
```

11.5.5 ManagerPresent action

```
acManagerPresent ACTION
  BEHAVIOUR
    bManagerPresent BEHAVIOUR
      DEFINED AS   !Receipt of this action may cause one or more Agent Present
                   notifications to be emitted, depending upon the evaluation of
                   the information held in the action arguments. The evaluation
                   is as follows:

                   -    The set of managed object instances that are eligible
                     to be included in the Agent Present notification, the
                     "eligible set," is determined, by evaluating the set
                     intersection between the set of managed object instances
                     that the Agent currently supports, the "supported set," and
                     the set of managed object instances that the Manager
                     requested to be included, the "requested set." The
                     determination of the membership of the requested set is as
                     follows:
                     1) If the Action request contained an empty Class-Instance
                     List, the requested set is equal to the supported set.
                     2) If the Action request contained any managed object
                     classes for which no instance names were specified, the
                     requested set includes all members of the supported set
                     whose managed object class matches those specified in the
                     request.
                     3) If the Action request contained any managed object
                     classes for which instance names were specified, the
                     requested set includes all members of the supported set
                     whose managed object class and instance names match those
                     specified in the request. If the class specified in the
                     request was the "actual class" managed object class name
                     defined in ISO/IEC 10165-4 : 1992, then the requested set
                     contains all members of the supported set whose instance
                     names match those that were specified for the "actual class"
                     entry in the request.

                   -    If a ResponseType of Respond if Event Reporting Enabled
                     was specified in the request, those members of the eligible
                     set that are not currently enabled for event reporting to
                     the originating Manager are excluded from the Eligible Set.

                   -    If a ResponseType of Respond if Event Reporting Disabled
                     was specified in the request, those members of the eligible
                     set that are currently enabled for event reporting to the
                     originating Manager are excluded from the Eligible Set.

                   -    If the resultant eligible set contains no members, no
                     further action is taken.

                   -    If the resultant eligible set contains members, one or
                     more Agent Present notifications are emitted, each conveying
                     a subset of the information requested and indicating that
                     the information was solicited. !
    ;
  ;
  WITH INFORMATION SYNTAX IEEE802-1-DEFDefinitions.ManagerPresent;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802-1B(10007) action(9)
              managerpresent(2)};
```

11.5.6 AgentPresent notification

```
nAgentPresent NOTIFICATION
  BEHAVIOUR
    bAgentPresent BEHAVIOUR
      DEFINED AS   !This notification is used to announce the presence of a set
                   of managed object classes/instances that are supported by the
                   Agent. It may be emitted in the following circumstances, in
                   accordance with Agent policy:

                     1) On Agent initialization or start-up, to inform Managers
                     of the presence of the Agent and the managed object classes/
                     instances supported;
                     2) On creation of new instances by the station, to inform
                     Managers of the existence of the new capability, or
                     following receipt of an Event Forwarding Disable action
                     which results in managed objects that no longer have event
                     reporting enabled to any (or sufficient) managers;

                     3) In response to a Manager Present action.

                   In cases 1 and 2, the "Unsolicited" variant of the information
                   syntax is used; in case 3, the "Solicited" variant of the
                   information syntax is used. In the "Unsolicited" variant, the
                   level of detail provided in the information syntax is a local
                   matter. In the "Solicited" variant (case 3), the level of
                   detail shall at least match that of the action to which it is
                   responding, as follows:
                   -    If an empty Class-Instance List was provided in the
                     action request, the level of detail provided by the Agent
                     is a local matter.
                   -    If the Class-Instance List contained managed object
                     classes with no accompanying instance names, then the Agent
                     Present notification shall at least include the class names
                     of any such classes for which instances appear in the
                     Eligible Set (as described in the behaviour of the Manager
                     Present action).
                   -    If the Class-Instance List contained instance names,
                     then the Agent Present notification shall include the class
                     and instance names of any such instances which appear in the
                     Eligible Set (as described in the behaviour of the Manager
                     Present action). !
    ;
  ;
  WITH INFORMATION SYNTAX IEEE802-1-DEFDefinitions.AgentPresent;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007)notification(10)
               agentpresent(0)};
```

11.5.7 EFDisable notification

```
nEFDisable NOTIFICATION
  BEHAVIOUR
    bDEFnDisable BEHAVIOUR
      DEFINED AS   !This notification is used to announce that a particular set
                   of managed object instances has been disabled with respect to
                   event reporting to the Manager stations to which the
                   notification is directed. This notification is used only when
                   the decision to disable event forwarding was made by the
                   Agent; i.e. the notification is not used following event
                   forwarding disablement that was requested via the EFDisable
                   action. !
    ;
  ;
  WITH INFORMATION SYNTAX IEEE802-1-DEFDefinitions.EFDisable;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007) notification(10)
               efdisable(1)};
```

11.6 Management information definitions for Extended Notification Type table

11.6.1 Extended Notification Type table entry managed object class

This managed object class extends the structure of the Notification Type table to include a Managed Object Instances field that permits the specification of managed object instance names as well as managed object class names. It is defined as a subclass of the Notification Type table entry managed object class; in a given implementation, the table may contain a mixture of entries of both types.

Entries in the table whose Managed Object Classes attribute contains a single managed object class may have a Managed Object Instances attribute that specifies a list of one or more instance names. If the Managed Object Classes attribute contains other than one managed object class, the Managed Object Instances attribute shall contain the empty set.

Support of this managed object class is optional.

```
oExtendedNotificationTypeTableEntry MANAGED OBJECT CLASS
   DERIVED FROM oNotificationTypeTableEntry;
   CHARACTERIZED BY
      pExtendedNotificationTypeTableEntry PACKAGE
         ATTRIBUTES    aManagedObjectInstances ;
      ;
   ;
REGISTERED AS  {iso(1)member-body(2)us(840)ieee802-1B(10007)managedObjectClass(3)
               extendednotification(6)};
```

11.6.2 ManagedObjectInstances attribute

```
aManagedObjectInstances ATTRIBUTE
   WITH ATTRIBUTE SYNTAX IEEE802-1-DEFDefinitions.ManagedObjectInstances ;
   MATCHES FOR EQUALITY;
   BEHAVIOUR
      bManagedObjectInstances BEHAVIOUR
         DEFINED AS    !This attribute defines the set of managed object instances
                       to which an Extended Notification Type table entry managed
                       object relates. If the Managed Object Classes attribute has a
                       value that contains other than a single managed object class,
                       this attribute value shall be the empty set. Otherwise, it may
                       contain zero or more instance names. If it contains no
                       instance names, its meaning is interpreted as "all instances
                       of the class." !
      ;
   ;
REGISTERED AS  {iso(1) member-body(2) us(840) ieee802-1B(10007) defedName(7)
               managedinstances(15)};
```

11.7 ASN.1 definitions

The following ASN.1 module contains the ASN.1 type definitions required by the management information definitions for Discovery and Event Forwarding Enable/Disable.

```
   IEEE802-1-DEFDefinitions{iso(1) member-body(2) us(840) ieee802-1B(10007)
               asn1Module(2) defdefinitions(2) version1(0)}

   DEFINITIONS ::=

   BEGIN

   IMPORTS

      ObjectClass,ObjectInstanceFROMCMIP-1{joint-iso-ccittms(9)cmip(1)modules(0)
      protocol(3)}
   ;  -- End of IMPORTS

   AgentPresent ::= CHOICE {
      solicited    [0] IMPLICIT SET OF ObjectClassInstance,
      unsolicited  [1] IMPLICIT SET OF ObjectClassInstance }

   DEFName ::= GraphicString {"DEF"}

   EFDisable ::= SET OF ObjectClassInstance

   EFEnable ::= SEQUENCE {
      cpeRetryTimeout [0] IMPLICIT REAL OPTIONAL,
      cpeRetryCounter [1] IMPLICIT INTEGER OPTIONAL,
      classInstance   [2] IMPLICIT SET OF ObjectClassInstance }

-- The encoding used for the cpeRetryTimeout REAL value shall be restricted
-- to use only the NR2 form, as defined in ISO 6093 : 1985.   The use of this
-- encoding within the Basic Encoding Rules is defined in ISO/IEC 8825 : 1990.


   EFEnableResponse ::= SEQUENCE {
      characterSet INTEGER,
      classOrCode  CHOICE {
         managedObjectGroup   [0] IMPLICIT SET OF MOGClass,
         systemEnableRtnCode  [1] IMPLICIT EnableReturnCode }}

   EnableReturnCode ::= INTEGER {
      enable                       (0),
      reenable                     (1),
      classNotSupported            (2),
      instanceNotSupported         (3),
      accessDenied                 (4),
      resourceLimitation           (5) }

   InstanceStructure ::= SEQUENCE {
      objectInstance          [0] ObjectInstance,
      instancesEnableRtnCode  [1] IMPLICIT EnableReturnCode }

   ManagedObjectInstances ::= SET OF ObjectInstance

   ManagerPresent ::= SEQUENCE {
      respondIf            RespondIf DEFAULT eventReportingEnabledOrDisabled,
      objectClassInstance  SET OF ObjectClassInstance }

   MOGClass ::= SEQUENCE {
      managedObjectClass   [0] ObjectClass,
      classOrCode     CHOICE {
         instanceStructure  [0] IMPLICIT SET OF InstanceStructure,
         classEnableRtnCode [1] IMPLICIT EnableReturnCode }}

   ObjectClassInstance ::= SEQUENCE {
      mclass          [0] ObjectClass,
      objectInstances [1] IMPLICIT ManagedObjectInstances OPTIONAL}
```

```
   RespondIf ::= INTEGER {
      eventReportingEnabled (0),
      eventReportingDisabled (1),
      eventReportingEnabledOrDisabled (2)}

END
```

12. Use of group addresses for LAN/MAN Management

The use of group CPE addresses as the destination address in CPE exchanges permits LAN/MAN Management protocol exchanges between a single originating station and all receiving stations reachable in the LAN/MAN environment which recognize that group CPE address and which are reachable in the LAN/MAN environment to which the originating station is attached (subject to the requirements of LMMP and CPE with respect to the use of group addressing). A group CPE address consists of an LSAP address, which comprises a group MAC address combined with the standard LLC address reserved for use by the LAN/MAN Management protocol (see 7.4).

NOTE—In general, the use of group addressing in CPE exchanges is possible only where the LMMP exchange being carried is of a nonconfirmed type and where the CPE exchanges used to carry the LMMP do not use the enhanced reliability QOS option.

Two well-known group MAC addresses have been allocated for use with the 802.1B LAN/MAN Management protocol. The first group MAC address provides a generic address for all Manager stations. Its value, represented using the hexadecimal notation described in Section 5, Universal Addresses and Protocol Identifiers, of IEEE Std 802-1990, is as follows:

01-80-C2-00-00-18

The second group MAC address provides a generic address for all Agent stations. Its value, represented using the same hexadecimal notation, is as follows:

01-80-C2-00-00-1A

Annexes

Annex A[1]
PICS proforma

(normative)

A.1 Introduction

The supplier of a protocol implementation that is claimed to conform to this International Standard shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

a) By the protocol implementor, as a checklist to reduce the risk of failure to conform to the standard through oversight.
b) By the supplier and acquirer—or potential acquirer—of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma.
c) By the user—or potential user—of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from an incompatible PICS).
d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.2 Abbreviations and special symbols

M	Mandatory
O	Optional
<item>:	conditional-item symbol, dependent upon the support marked for <item>: (see A.3.4)

A.3 Instructions for completing the PICS proforma

A.3.1 General structure of the PICS proforma

The first part of the PICS proforma—Identification, A.4—is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire divided into four subclauses containing groups of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No), or by entering a value or a set or range of values.

[1]*Copyright release for PICS proformas:* Users of this International Standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

Each item is identified by an item reference in the first column; the second column contains the question to be answered; and the third column contains the reference or references to the material that specifies the item in the main body of ISO/IEC 15802-2 : 1995. The remaining columns record the status of the item—whether support is mandatory or optional—and provide the space for answers (see also A.3.4).

A supplier may also provide, or can be required to provide, further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labeled A*i* or X*i*, respectively, for cross-referencing purposes, where *i* is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format and presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the Protocol Implementation Conformance Statement for the implementation in question.

NOTE: Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's configuration capabilities, in case that makes for easier and clearer presentation of the information.

A.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. An example might be an outline of the ways in which a single implementation can be set up to operate in a variety of environments and configurations.

References to items of Additional Information may be entered next to any answers in the questionnaire, and may be included in items of Exception Information.

A.3.3 Exception information

It may occasionally happen that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer will be found in the Support column for this; instead, the supplier shall write the missing answer in the Support column, together with an X*i* reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to ISO/IEC 15802-2 : 1995.

NOTE—A possible reason for the situation described above is that a defect in this International Standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.3.4 Conditional items

The PICS proforma contains a number of conditional items. These are items for which the applicability of the item itself, and its status if it does apply—mandatory or optional—are dependent upon whether or not certain other items are supported.

Individual conditional items are indicated by a conditional symbol of the form "<item>: S", where <item> is an item reference that appears in the first column of the table for some other item, and S is one of the status symbols M or O.

If the item referred to by the conditional symbol is marked as supported, the conditional item is applicable, and its status is given by S; the support column is marked in the usual way. Otherwise, the conditional item is not relevant, and the Not Applicable (N/A) answer is to be marked.

Each item whose reference is used in a conditional symbol is indicated by an asterisk in the Item column.

A.4 Identification

A.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machine(s) and/or operating systems; System Name(s)	

NOTES

1—Only the first three items are required for all implementations; the other information may be completed as appropriate in meeting the requirements for full identification.

2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).

A.4.2 Protocol summary, IEEE 802.1B convergence protocol

Identification of protocol specification	ISO/IEC 15802-2 : 1995
Identification of amendments and corrigenda to this PICS proforma which have been completed as part of this PICS	ISO/IEC 15802-2 : 1995: Amd. : Corr. : Amd. : Corr. :
Have any Exception items been required (see A.3.3)? (The answer Yes means that the implementation does not conform to ISO/IEC 15802-2 : 1995.)	No [ ] Yes [ ]
Date of Statement	

A.5 Major capabilities

Item	Feature	References	Status	Support
	Does the implementation support information exchange by:			
CPur	—Reception of unconfirmed requests (basic Reliability QOS)?	7.3, 10.1.2 (b1)	M	Yes [ ]
CPug	—Generation of unconfirmed requests (basic Reliability QOS)?	7.3, 10.1.2 (b3)	M	Yes [ ]
CPcr	—Reception of, and response to, confirmed requests (enhanced Reliability QOS)?	7.3, 10.1.2 (b2)	M	Yes [ ]
CPcg	—Generation of confirmed requests, and reception of responses (enhanced Reliability QOS)?	7.3, 10.1.2 (b4)	M	Yes [ ]
CPrg	Does the implementation support request groups of size greater than 1?	7.3.1.3, 10.1.4	O	Yes [ ] No [ ]
CPms	Does the implementation support the reception of CPDUs of up to and including 1500 octets in length?	7.3.6.2, Table 7-5	M	Yes [ ]
MOlm	Does the implementation support management of the operation of the Convergence Protocol via the LAN/MAN Management managed object?	8.2, 10.1.3	M	Yes [ ]
MOsi	Does the implementation support management of specific CPE timer and counter information via the Specific CPE Info managed object?	8.3, 10.1.4	O	Yes [ ] No [ ]
MOrt	Does the implementation support identification of manufacturer and product version information via the Resource Type ID managed object?	8.4, 10.1.3	M	Yes [ ]
*MOac	Does the implementation support the management of access control table information via the Access Class Table Entry managed object?	8.5, 9.2, 10.1.4	O	Yes [ ] No [ ]
*MOev	Does the implementation support the management of notification type table information via the Notification Type Table Entry and Event Report Destination Table Entry managed objects?	8.6, 8.7, 9.1, 10.1.4	O	Yes [ ] No [ ]
MOdef	Does the implementation support the Discovery and Event Forwarding Enable/Disable mechanisms?	10.1.4, 11	O	Yes [ ] No [ ]
*MOent	Does the implementation support the management of notification type table information via the Extended Notification Type Table Entry managed object?	10.1.4, 11.6	O	Yes [ ] No [ ]

A.6 Convergence protocol details

Item	Feature	References	Status	Support
CPpdu	Does the implementation support the specified CPDU structure and encoding?	7.3.6, 10.1.2 (b5)	M	Yes [ ]
*CPllc	Does the implementation support the exchange of CPDUs by means of LLC Type 1 procedures?	7.4, 10.1.2 (c)	M	Yes [ ]

A.7 Convergence protocol parameters

Item	Feature	References	Status	Support
	State the values, or ranges of values,supported by the implementation for:			
CPVgs	—Max group size	7.3.2.1	M	Size ___
CPVrt	—Retry limit (range)	7.3.2.2	M	From __ to __
CPVto	—Timeout (range), in seconds	7.3.2.2	M	From __ to __
CPVna	—Maximum number of affiliations (affiliate records) that can be supported simultaneously	7.3.2.2	M	Max __
CPVla	—Minimum and maximum lifetime of an affiliate record after the last communication with the affiliate, in seconds	7.3.2.2	M	Min __ Max __
CPVnr	—Maximum number of outstanding requests (request records), across all request groups, that the implementation can support simultaneously	7.3.2.3	M	Max __
CPVrp	—Maximum size (in octets) of received CPDUs	7.3.6.2, Table 7-5	M	Max __
CPVtp	—Maximum size (in octets) of transmitted CPDUs	7.3.6.2, Table 7-5	M	Max __

A.8 Managed object support

Item	Feature	References	Status	Support
	State the values supported by the implementation for:			
CIVne	—Maximum number of Specific CPE Info managed objects	8.3	MOsi:M	N/A [ ] Max __
ACVne	—Maximum number of Access Class Table Entry managed objects	8.5	MOac:M	N/A [ ] Max __
ECVne	—Maximum number of Notification Type Table Entry managed objects	8.6	MOev:M	N/A [ ] Max __
EDVne	—Maximum number of Event Report Destination Table Entry managed objects	8.7	MOev:M	N/A [ ] Max __
	State the values supported by the implementation for:			
EENTne	—Maximum number of Extended Notification Type table entry managed objects	11.6	MOent: M	N/A [] Max __

Annex B
Allocation of object identifier values

(normative)

B.1 Introduction

This annex contains a summary of all object identifier values that have been allocated by this International Standard, both in this revision and in previous revisions.

Each table shows allocations related to a specific category of information object. The heading of the table identifies the category of information object, and shows the invariant part of the object identifier value allocated to the entries in the table. The column marked Arc shows the value allocated to the arc subsequent to the invariant part, which completes the object identifier value allocated. The column marked Purpose contains a text description of the information object, and, in the case of current allocations, a reference to the location of the definition of the information object in the standard. The columns marked Status show the status of the allocated values, using the following convention:

R Reserved. The object identifier value is reserved for future use by this standard.

C Current. The object identifier value has been allocated to an information object that is defined within the current revision of the standard.

D Deprecated. The object identifier value has been allocated to an information object that was defined in a previous revision of the standard, and whose use is now deprecated.

B.2 Allocation tables

Allocations for standard-specific extensions.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) standardSpecificExtensions(0)}

ARC	PURPOSE	STATUS
None allocated	N/A	N/A

Allocations for ASN.1 module identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) asn1Module(2)}

ARC	PURPOSE	STATUS
convergenceprotocol(0) version1(0)	Version 1 of the CPDU definition module [7.3.6.3]	C
lmmdefinitions(1) version1(0)	Version 1 of the LMM managed object definition module [8.8.2]	C
defdefinitions(2) version1(0)	Version 1 of the DEFED managed object definition module [11.7]	C

Allocations for Managed object classes.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) managedObjectClass(3)}

ARC	PURPOSE	STATUS
lmmclass(0)	LAN/MAN Management managed object class name [8.2]	C
specificCPEinfo(1)	Specific CPE Info managed object class name [8.3]	C
accessentry(2)	Access Class Table Entry managed object class name [8.5]	C
notificationentry(3)	Notification Type Table Entry managed object class name [8.6]	C
eventdestentry(4)	Event Report Destination Table Entry managed object class name [8.7]	C
defedMO(5)	Discovery and Event Forwarding Enable/Disable managed object class name [11.5]	C
extendednotification(6)	Extended Notification Type Table Entry managed object class name [11.6.1]	C

Allocations for Package identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) packages(4)}

ARC	PURPOSE	STATUS
qospackage(0)	QOS Conditional Package identifier [8.7]	C

Allocations for Parameters.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) parameter(5)}}

ARC	PURPOSE	STATUS
None Allocated	N/A	N/A

Allocations for Name Binding identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) nameBinding(6)}

ARC	PURPOSE	STATUS
lmmbinding(0)	LAN/MAN Management managed object name binding [8.2]	C
specificCPEbinding(1)	Specific CPE Info managed object name binding [8.3]	C
resIDbinding(2)	Resource Type ID managed object name binding [8.4]	C
atebinding(3)	Access Class Table Entry managed object name binding [8.5]	C
notificationentrybinding(4)	Notification Type Table Entry managed object name binding [8.6]	C
eventdestentrybinding(5)	Event Report Destination Table Entry managed object class name [8.7]	C
defednamebinding(6)	DEFED managed object name binding [11.5]	C

Allocations for Attribute identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) attribute(7)}

ARC	PURPOSE	STATUS
lmmname(0)	LMM Name attribute [8.2.1]	C
defaultCPEinfo(1)	Default CPE Info attribute [8.2.2]	C
cpeaddress(2)	CPE Address attribute [8.3.1]	C
atename(3)	Access Class Table Entry Name attribute [8.5.1]	C
moclass(4)	Managed Object Classes attribute [8.5.2]	C
password(5)	Password attribute [8.5.3]	C
localclasses(6)	Local Access Classes attribute [8.5.4]	C
remoteclasses(7)	Remote Access Classes attribute [8.5.5]	C
notypename(8)	Notification Type Table Entry Name attribute [8.6.1]	C
notypes(9)	Notification Types attribute [8.6.2]	C
evendests(10)	Event Report Destinations attribute [8.6.3]	C
edename(11)	Event Report Destination Table Entry Name attribute [8.7.1]	C
ede(12)	Event Report Destination Address attribute [8.7.2]	C
qos(13)	Destination QOS attribute [8.7.3]	C
defname(14)	DEF Name attribute [11.5.2]	C
managedinstances(15)	Managed Object Instances attribute [11.6.2]	C

Allocations for Attribute Group identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) attributeGroup(8)}

ARC	PURPOSE	STATUS
None allocated	N/A	N/A

Allocations for Action types.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) action(9)}

ARC	PURPOSE	STATUS
efenable(0)	DEF Enable action [11.5.3]	C
efdisableAction(1)	DEF Disable action [11.5.4]	C
managerpresent(2)	DEF Manager Present action [11.5.5]	C

Allocations for Notification types.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802-1B(10007) notification(10)}}

ARC	PURPOSE	STATUS
agentpresent(0)	DEF Agent Present notification [11.5.6]	C
efdisable(1)	DEF Disable notification [11.5.7]	C