INTERNATIONAL STANDARD

ISO/IEC/IEEE 8802-21

First edition
2018-04

Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements —

Part 21: Media independent services framework

Technologies de l'information — Télécommunications et échange d'information entre systèmes — Réseaux locaux et métropolitains — Exigences spécifiques —

Partie 21: Cadre des services indépendants des supports

Reference number
ISO/IEC/IEEE 8802-21:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

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

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted.

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

ISO/IEC/IEEE 8802-21 was prepared by the LAN/MAN of the IEEE Computer Society (as IEEE Std 802.21-2017) and drafted in accordance with its editorial rules. It was adopted under the "fast-track procedure" defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems.*

A list of all parts in the ISO/IEC/IEEE 8802 series can be found on the ISO website.

IEEE Std 802.21™-2017
(Revision of IEEE Std 802.21-2008
as amended by IEEE Std 802.21a™-2012,
IEEE Std 802.21b™-2012, IEEE Std 802.21c™-2014,
and IEEE Std 802.21d™-2015)

IEEE Standard for Local and metropolitan area networks—

Part 21: Media Independent Services Framework

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 14 February 2017

IEEE-SA Standards Board

Abstract: An extensible IEEE 802® media access independent services framework (i.e., function and protocol) is defined that enables the optimization of services including handover and other services when performed between heterogeneous IEEE 802 networks. These services are facilitated by this standard when networking between IEEE 802 networks and cellular networks.

Keywords: broadcast, downlink only, group, group management, group security, IEEE 802.21™, management, media independent handover, media independent service, mobile node, mobility, multicast, point of attachment, point of service, proactive authentication, seamless, security protection, service access authentication

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

PDF: ISBN 978-1-5044-3806-3 STD22456
Print: ISBN 978-1-5044-3807-0 STDPD22456

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

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.

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

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

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

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Xplore at http://ieeexplore.ieee.org/ or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the P802.21 Working Group had the following membership:

Subir Das, *Chair*
Hyeong Ho Lee, *Vice Chair*
Yoshikazu Hanatani, *Technical Editor*

H. Anthony Chan
Clint Chaplin
Lidong Chen
Jin Seek Choi
Daniel Corujo
Antonio De la Oliva Delgado
Yong-Geun Hong
Sangkwon Peter Jeong
Farrokh Khatibi
Michael Lynch
Yoichi Masuda
Naoki Ogura
Yoshihiro Ohba
Hyunho Park
Charles E. Perkins
Karen Randall
Yusuke Shimizu
Tomoki Takazoe
Keiichi Teramoto
Yuji Unagami

In addition, the following members have either contributed or participated during the development of this Standard:

Yusuke Doi
Krzysztof Grochla
Changhwa Lyou
Torleiv Masen
Christian Niephaus
Dick Roy
Ruben Salazar
Randy Turner

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

Thomas Alexander
Butch Anton
H. Stephen Berger
Harry Bims
Gennaro Boggia
William Byrd
Juan Carreon
Charles Cook
Daniel Corujo
Subir Das
Sourav Dutta
Richard Edgar
Marc Emmelmann
Avraham Freedman
Joel Goergen
Randall Groves
Yoshikazu Hanatani
Werner Hoelzl
David Howard
Noriyuki Ikeuchi
Atsushi Ito
Raj Jain
Piotr Karocki
Stuart Kerry
Farrokh Khatibi
Yongbum Kim
Hyeong Ho Lee
Jae Seung Lee
Moon-Sik Lee
Michael Lynch
Elvis Maculuba
Stephen McCann
Michael McInnis
Jeffrey Moore
Nick S. A. Nikjoo
Paul Nikolich
Yoshihiro Ohba
Satoshi Oyama
Arumugam Paventhan
Venkatesha Prasad
Karen Randall
Maximilian Riegel
Naotaka Sato
Yusuke Shimizu
Dorothy Stanley
Thomas Starai
Michael Stelts
Walter Struppler
Mark Sturza
Tomoki Takazoe
Patricia Thaler
Mark-Rene Uchida
Dmitri Varsanofiev
Prabodh Varshney
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 14 February 2017, it had the following membership:

Jean-Philippe Faure, *Chair*
Vacant Position, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Chuck Adams
Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Gary Hoffman
Michael Janezic
Thomas Koshy
Joseph L. Koepfinger*
Kevin Lu
Daleep Mohla
Damir Novosel
Ronald C. Petersen
Annette D. Reilly
Robby Robson
Dorothy Stanley
Adrian Stephens
Mehmet Ulema
Phil Wennblom
Howard Wolfman
Yu Yuan

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 802.21-2017, IEEE Standard for Local and metropolitan area networks—Part 21: Media Independent Services Framework.

This standard defines an extensible IEEE 802® media access independent services framework (i.e., function and protocol) that enables the optimization of services including handover service when performed between heterogeneous IEEE 802 networks. It also facilitates these services when networking between IEEE 802 networks and cellular networks.

Contents

List of Figures

List of Tables

IEEE Standard for
Local and metropolitan area networks—

Part 21: Media Independent Services Framework

1. Overview

1.1 Scope

This standard defines an extensible IEEE 802® media access independent services framework (i.e., function and protocol) that enables the optimization of services including handover service when performed between heterogeneous IEEE 802 networks. It also facilitates these services when networking between IEEE 802 networks and cellular networks.

1.2 Purpose

The purpose of this standard is to improve the user experience of mobile devices by describing a framework and knobs that several services can utilize in a media independent manner, including the handover service between heterogeneous IEEE 802 networks. This framework is also applicable for interworking between IEEE 802 networks and cellular networks.

1.3 General

This standard provides link-layer intelligence and other related network information to upper layers of a mobile device or a network element to support several use cases, such as handovers between heterogeneous networks, radio resource management, home energy management, software-defined radio access networks, and device-to-device (D2D) communication as described in IEEE Std 802.21.1™-2017. In this standard, unless otherwise noted, *media* refers to the method/mode of accessing a telecommunication system (e.g., cable, radio, satellite), as opposed to sensory aspects of communication (e.g., audio, video).

The following items are not within the scope of this standard:

— Enhancements specific to particular link-layer technologies that are required to support this standard (they should be carried out by those respective link-layer technology standards)

— Media-specific protection mechanisms

— Higher layer (layer 3 and above) enhancements that are required to support this standard

The purpose of this standard is to provide a framework with several knobs so that they can be utilized to enhance the experience of mobile users while they are performing functions, such as handovers between heterogeneous networks when mobile, managing link-layer radio resources with or without presence of

software-defined networking, and obtaining group keys for home energy-management systems via multicast group management.

This standard supports another important aspect of optimized performance enhancement through link adaptation. For example, a user chooses an application that requires a higher data rate than available on the current link, necessitating a link adaptation to provide the higher rate, or necessitating an action if the higher rate is unavailable on the current link. In all such cases, service continuity and/or user experience should be maintained to the extent possible during this action. As an example, when making a network transition during a phone call, the handover procedures should be executed in such a way that any perceptible interruption to the conversation should be minimized.

This standard supports cooperative use of information available at the mobile node and within the network infrastructure. The mobile node is well-placed to detect available network resources based on the use cases that they are performing. The network infrastructure is well-suited to store the necessary information that is required to provide either a better user experience or managing the mobile devices better. The information could be related to handover and radio-resource management, such as neighborhood cell lists, location of mobile nodes, available link-layer radio resources and higher layer service availability, home energy-management system such as multicast group information with their keys, and certificates.

The overall network includes a mixture of cells of drastically different sizes, such as those from IEEE Std 802.15™, IEEE Std 802.11™, IEEE Std 802.16™, 3GPP™[1], and 3GPP2, with overlapping coverage. The specific use case is initiated either by the mobile node or by a network node. They could be specific measurement reports, triggers supplied by the link layers, unavailability of a key or a certificate. Specifically the standard consists of the following elements:

a) A framework that enables the optimization of handover and other services supporting several use cases described in IEEE Std 802.21.1-2017. The framework relies on the presence of a higher layer applications such as mobility-management protocol stack within the network elements that supports the handover, and a group manager function that manages groups of mobile nodes and distributes the keys and certificates. The framework presents media independent service (MIS) reference models for different link-layer technologies so that all actions from the higher layer can be performed with minimum or no modifications of link-layer technologies.

b) A set of media independent functions within the protocol stacks of the network elements and a new entity created therein called the MIS function (MISF).

c) A media independent service access point (called the MIS_SAP) and associated primitives are defined to provide MIS users with access to the services of the MISF. The MISF provides the following services:

   1) The media independent event service that detects changes in link-layer properties and initiates appropriate events (triggers) from both local and remote interfaces.

   2) The media independent command service provides a set of commands for the MIS users to control link properties that are relevant to handover and other services.

   3) The media independent information service provides the information about different networks and their services, thus enabling more effective handover and other management decisions to be made across heterogeneous networks.

d) Media independent protocol messages and their protection mechanisms using both unicast and multicast modes of transmission.

e) The definition of new link-layer service access points (SAPs) and associated primitives for each link-layer technology as applicable to handover and other use cases described in IEEE Std 802.21.1-2017. The new primitives help the MISF collect link information and control link behavior during handovers.

[1] 3GPP is a trademark of The European Telecommunications Standards Institute (ETSI).

Figure 1 shows the placement of the MISF within the protocol stack of a multiple interfaced mobile node (MN) or network entity. The MISF provides services to the MIS users through a single media independent interface (the MIS service access point) and obtains services from the lower layers through a variety of media dependent interfaces (media-specific SAPs).

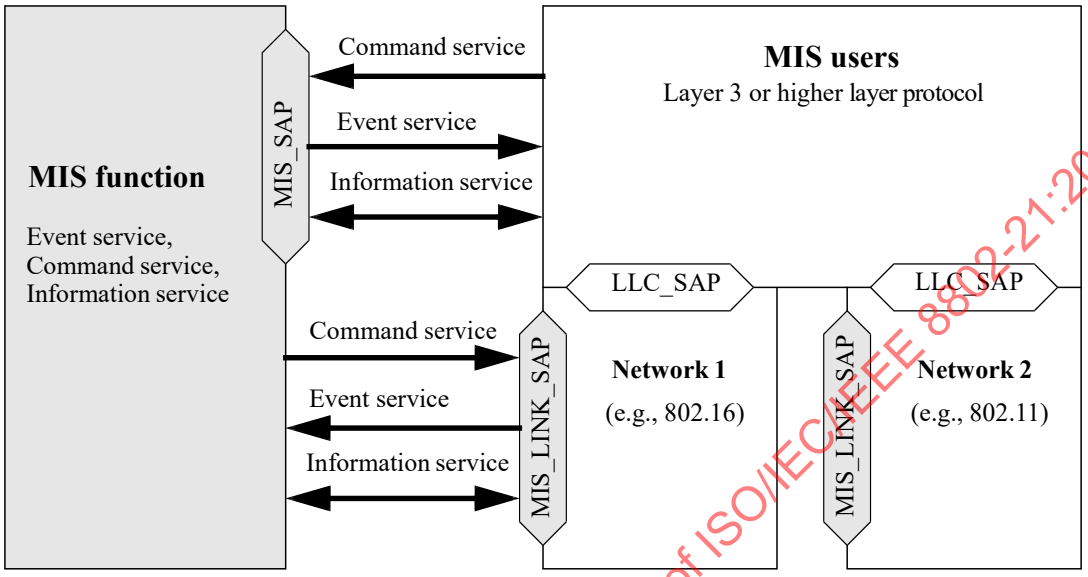


Figure 1—MIS services and their initiations

1.4 Assumptions

The following assumptions have been made in the development of this standard:

a) The MN is capable of supporting multiple link-layer technologies, such as wireless, wired, or mixed.

b) The MISF is a logical entity, whose definition is independent of its deployment location on the MN or in the network.

c) The MISF, regardless of whether it is located on the MN or in the network, receives and transmits information about the configuration and condition of access networks around the MN. This information originates at different layers of the protocol stack within the MN or at various network elements.

   1) When the information originates at a remote network element, the MISF on the local network element obtains it through MIS message exchanges with a peer MISF instance that resides in the remote network element.

   2) When the information originates at lower layers of the protocol stack within an MN or network entity, the MISF on that entity obtains it locally through the service primitives of the SAPs that define the interface of the MISF with the lower layers.

1.5 Media independence

The intent of this standard is to provide generic link-layer intelligence and other network resources information independent of the specifics of mobile nodes or radio networks.

The defined SAPs and primitives in this standard provide generic link-layer intelligence. Depending upon the specific use cases, individual media-specific technologies may need to be enhanced to support the media-specific SAPs and primitives and to satisfy the generic abstractions of this standard.

2. Normative references

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

3GPP TS 23.003 (2015-06), Numbering, addressing and identification (Release 12).

3GPP TS 25.215 (2015-03), Physical layer—Measurements (FDD) (Release 12).

3GPP TS 25.401 (2013-12), UTRAN overall description (Release 12).

3GPP TS 25.413 (2015-03), UTRAN Iu interface Radio Access Network Application Part (RANAP) signalling (Release 12).

3GPP TS 45.008 (2014-09), Radio subsystem link control (Release 12).

3GPP2 C.S0004-D (2004-03), Signaling Link Access Control (LAC) Standard for cdma2000 Spread Spectrum Systems.

ANSI X3.159-1989: Programming Language C.[2]

FIPS 198, The Keyed-Hash Message Authentication Code (HMAC).[3]

IEEE Std 802.1AB™-2009, IEEE Standard for Local and Metropolitan Area Networks—Station and Media Access Control Connectivity Discover. [4, 5]

IEEE Std 802.1AR™-2009, IEEE Standard for Local and Metropolitan Area Networks: Secure Device Identity.

IEEE Std 802.1Q™-2014, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks.

IEEE Std 802.1X™-2010, IEEE Standard for Local and metropolitan area networks—Port-Based Network Access Control.

IEEE Std 802.3™-2012, IEEE Standard for Ethernet.

IEEE Std 802.11™-2012, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

IEEE Std 802.16™-2012, IEEE Standard for Air Interface for Broadband Wireless Access Systems.

[2] ANSI publications are available from the American National Standards Institute (http://www.ansi.org/).

[3] FIPS publications are made available at http://csrc.nist.gov/publications/PubsFIPS.html.

[4] IEEE publications are available from the Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

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

IEEE Std 802.16.1™-2012, IEEE Standard for WirelessMAN-Advanced Air Interface for Broadband Wireless Access Systems.

IEEE Std 802.20™-2008, IEEE Standard for Local and metropolitan area networks—Part 20: Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility—Physical and Media Access Control Layer Specification.

IEEE Std 802.21.1™-2017, IEEE Standard for Standard for Local and metropolitan area networks Part 21.1: Media Independent Services.

IEEE Std 802.22™-2011, IEEE Standard for Information Technology—Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN)—Specific requirements—Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands.

IETF RFC 1661 (1994-07), The Point-to-Point Protocol (PPP).[6]

IETF RFC 2627 (1999-06), Key Management for Multicast: Issues and Architectures.

IETF RFC 2865 (2000-06), Remote Authentication Dial In User Service (RADIUS).

IETF RFC 2988 (2000-11), Computing TCP's Retransmission Timer.

IETF RFC 3748 (2004-06), Extensible Authentication Protocol (EAP).

IETF RFC 4119 (2005-12), A Presence-based GEOPRIV Location Object Format.

IETF RFC 4302 (2005-12), IP Authentication Header.

IETF RFC 4303 (2005-12), IP Encapsulating Security Payload (ESP).

IETF RFC 4443 (2006-03), Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.

IETF RFC 4555 (2006-06), IKEv2 Mobility and Multihoming Protocol (MOBIKE).

IETF RFC 4776 (2006-11), Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information.

IETF RFC 5184 (2008-05), Unified L2 Abstractions for L3-Driven Fast Handover.

IETF RFC 5246 (2008-08), The Transport Layer Security (TLS) Protocol Version 1.2.

IETF RFC 5280 (2008-05), Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.

IETF RFC 5677 (2009-12), IEEE 802.21 Mobility Services Framework Design (MSFD).

IETF RFC 5678 (2009-12), Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Options for IEEE 802.21 Mobility Server (MoS) Discovery.

IETF RFC 5679 (2009-12), Locating IEEE 802.21 Mobility Services Using DNS.

[6] IETF RFCs are available from the Internet Engineering Task Force website at http://www.ietf.org/rfc.html.

IETF RFC 6225 (2011-07), Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information.

IETF RFC 6347 (2012-01), Datagram Transport Layer Security Version 1.2.

IETF RFC 6733 (2012-10), Diameter Base Protocol.

IETF RFC 6696 (2008-08), EAP Extensions for EAP Re-authentication Protocol (ERP).

IETF RFC 7296 (2014-06), Internet Key Exchange Protocol Version 2 (IKEv2).

IETF RFC 7542 (2015-05), The Network Access Identifier.

ISO 3166-1 (1997), Codes for the representation of names of countries and their subdivisions—Part 1: Country codes.[7]

ISO 4217, Codes for the Representation of Names of Countries.

ISO/IEC 8802-2:1998, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

ITU-T Recommendation X.290 (1995), OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications—General concepts.[8]

ITU-T Recommendation X.296 (1995), OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications—Implementation conformance statements.

ITU-T Recommendation Y.1540, Internet protocol data communication service—IP packet transfer and availability performance parameters.

NIST SP 800-38A, Recommendation for Block Cipher Modes of Operation.[9]

NIST SP 800-38B, Recommendation for Block Cipher Modes of Operation—The CMAC Mode for Authentication.

NIST SP 800-38C, Recommendation for Block Cipher Modes of Operation—The CCM Mode for Confidentiality and Authentication.

NIST SP 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, 2012.

NIST SP800-108, Recommendation for Key Derivation using Pseudorandom Functions, 2009.

W3C Recommendation, RDF/XML Syntax Specification.[10]

W3C Recommendation, Resource Description Framework (RDF)—Concepts and Abstract Syntax.

W3C Recommendation, SPARQL Query Language for RDF.

[7] ISO publications are available from the ISO Central Secretariat (http://www.iso.org/). ISO publications are also available from the American National Standards Institute (http://www.ansi.org/).
[8] ITU-T publications are available from the International Telecommunications Union (http://www.itu.int/).
[9] NIST publications are available from the National Institute of Standards and Technology (http://www.nist.gov/).
[10] W3C recommendations are available from http://www.w3.org.

3. Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[11]

authenticated encryption: An algorithm to convert plaintext data to ciphertext and generate a message authentication code with a cryptographic key as a parameter to provide confidentiality, integrity, and authenticity of the data. *See also:* **encryption**; **MIC algorithm**.

authentication process: A process to assure that the claimed identity belongs to the entity. It is also called entity authentication. In this standard, an access authentication is an entity authentication with the identity used to access a specific network or a media independent service (MIS).

authentication server: A server used for authentication purposes. When extensible authentication protocol (EAP) is used as an authentication protocol, the authentication server is an EAP server.

authenticator: A network entity to execute extensible authentication protocol (EAP) with a mobile node called a peer. An authenticator can use a backend server to conduct EAP execution. *Syn*: **EAP authenticator**.

candidate network: A network that is a potential target to the mobile node's movement.

candidate point of attachment (candidate PoA): A point of attachment (PoA) under evaluation to which the link may be switched.

certificate authority (CA): A trusted entity that issues and revokes public key certificates.

decryption: An algorithm to convert ciphertext of data to plaintext with a cryptographic key as a parameter. It is an inverse operation of encryption.

device key: A cryptographic key assigned to a device. Each device is provisioned with a set of device keys.

EAP authenticator: *See:* **authenticator**.

EAP peer: The entity that responds to the extensible authentication protocol (EAP) authenticator.

EAP re-authentication: An authentication protocol using a key established in a previous extensible authentication protocol (EAP) execution as defined in IETF RFC 6696.

EAP server: The entity that terminates the extensible authentication protocol (EAP) execution with the EAP peer. In the case where no backend authentication server is used, the EAP server is a part of the EAP authenticator. In the case where a backend authentication server is used, the EAP server is located on the backend authentication server.

encryption: An algorithm to convert plaintext data to ciphertext to provide confidentiality with a cryptographic key as a parameter.

extensible authentication protocol (EAP): An access authentication framework specified in IETF RFC 3748. It supports different authentication methods, called EAP methods.

group addressed message: A media independent service (MIS) message sent to nodes that belongs to a group identified by an MIS function (MISF) Group ID. A group addressed message is sent using a multicast transport mechanism, which is out of the scope of this specification. An exception to this is that a unicast transport mechanism when used for a two-member group.

group key block (GKB): A data entity carrying an encapsulated group key in such a manner that only those nodes that have the corresponding device keys are able to decapsulate it.

group key wrapping: Security procedure that uses group key block (GKB) as a data element to encrypt group keys.

[11] *IEEE Standards Dictionary Online* is available at: http://dictionary.ieee.org/.

group management tree: A perfect binary tree used for group management purposes.

group manager (GM): A media independent service (MIS) user that manages the group by adding, deleting, and updating the group membership information. It also generates the group key when needed. It is also the entity that issues group manipulation commands.

group manipulation command: A command, sent to a group of nodes or to an individual node, that instructs the recipient to perform certain operations such as joining and leaving a group, updating group membership, and so on. By group manipulation command, we refer to the following commands: MIS_Pull_Group_Manipulate and MIS_Push_Group_Manipulate.

handover policies: A set of rules that contribute to making the handover decision for a mobile node.

handover: The process by which a mobile node obtains facilities and preserves traffic flows upon occurrence of a link switch event. The mechanisms and protocol layers involved in the handover vary with the type of the link switch event (i.e., with the type of the serving and target point of attachment and the respective subnet associations). Different types of handover are defined based on the way facilities for supporting traffic flows are preserved. *See also:* **hard handover**; **soft handover**; **seamless handover**.

hard handover: Handover where facilities for supporting traffic flows are subject to complete unavailability between their disruption on the serving link and their restoration on the target link (break-before-make).

horizontal handovers: A handover where a mobile node moves between point of attachments of the same link type (in terms of coverage, data rate, and mobility), such as universal mobile telecommunications systems (UMTS™[12]) to UMTS or wireless local area network (WLAN) to WLAN. *Syn:* **intra-technology handovers**.

information server: A server providing information about candidate access networks. The information server may be implemented in a media independent information server but may also be implemented with other standards such as the Access Network Discovery and Selection Function (ANDSF) defined in 3GPP or a server using Access Network Query Protocol (ANQP) defined in IEEE Std 802.11-2012.

internal node: A node having two children and one parent node.

inter-technology handovers: *See:* **vertical handovers**.

leaf key: A node key which is assigned to a leaf node of a group management tree.

leaf node: A node having one parent node without having a child node.

leaf number: The integer representation of the node index assigned to a leaf node.

link layer: Conceptual layer of control or processing logic that is responsible for maintaining control of the data link. The data link-layer functions provide an interface between the higher layer logic and the data link.

link switch: The process by which a mobile node changes the link that connects it to the network. Changing a link implies changing the remote link endpoint and therefore the point of attachment of the mobile node.

link: A communication channel through which nodes communicate for the exchange of L2 protocol data units. Each link is associated with two endpoints and has a unique identifier.

lower layers: The layers located at OSI Level 2 and below across different link-layer technology standards supported by this standard. For example, the IEEE 802.11 lower layers are the MAC sublayer and the PHY, while the 3GPP lower layers are L1/MAC/radio link control (RLC)/packet data convergence protocol (PDCP) in the case of wideband code division multiple access (W-CDMA) frequency division duplex (FDD)/time division duplex (TDD), respectively. The term *lower layers* also includes logical link control (LLC) layers such as IEEE 802.2 LLC or 3GPP radio link control (RLC). The MISF uses the services provided by these layers.

[12] UMTS is a trademark of The European Telecommunications Standards Institute (ETSI).

media independent service (MIS) network entity: Network entity with media independent service function (MISF) capability.

media independent service (MIS) node: An media independent service function (MISF) capable entity (mobile node or network).

media independent service (MIS) non-PoS: An MIS network entity that directly exchanges MIS messages with other MIS network entities but is not capable of directly exchanging MIS messages with any MIS-enabled mobile node.

media independent service (MIS) transport protocol: A protocol for transporting MIS protocol messages between a pair of MIS entities.

media independent service (MIS) users: Entities that use the services provided by the MISF. MIS users use the MIS_SAP to interact with the MISF.

media independent service function (MISF): A function that realizes MIS services.

media independent service function broadcast identifier (MISF Broadcast ID): An MISF Group ID of zero length, i.e., '0x00' (OCTET_STRING with *Length* field = 0x00 and no *Value* field).

media independent service function group identifier (MISF Group ID): An identifier of a group of MISF peer entities.

media independent service point of service (MIS PoS): Network-side MISF instance that exchanges MIS messages with an MN-based MISF. The same MIS network entity includes an MIS PoS for each MIS-enabled mobile node with which it exchanges MIS messages. A single MIS PoS is capable of hosting more than one MIS service. The MIS network entity that includes multiple MIS Points of Service is capable of providing different combinations of MIS services to the respective mobile nodes based on subscription or roaming conditions. Note that for a network entity comprising multiple interfaces, the notion of MIS PoS is associated with the network entity itself and not with just one of its interfaces. For MIS service access authentication, a PoS serves as an authenticator. Moreover, when a service access authentication establishes keys for proactive authentication, a PoS provides key distribution service for media-specific authenticators.

media-specific authenticator: An authenticator used for a media-specific network access authentication.

media-specific network access authentication: An authentication protocol for media access purpose specified for a specific media access. It establishes keys to be used in media-specific protection mechanisms.

media-specific protection mechanism: A mechanism that is applied to media-specific layers to protect the data traffic using an encryption algorithm, an integrity protection algorithm, an authenticated encryption algorithm, or a combination of an encryption algorithm and an integrity protection algorithm.

message authentication code: A data string generated over a message with a symmetric key by an algorithm, called *message authentication code algorithm*. It is used to verify the integrity of the message and to authenticate the origin of the message. *Syn:* **message integrity code**.

message authentication code algorithm: An algorithm to generate a message authentication code on a data message with a symmetric key to provide integrity protection and message origination authentication. *See also:* **message authentication code**.

message integrity code (MIC): *See:* **message authentication code**.

MIS security association (SA): A media independent service (MIS) security association is a set of cryptographic attributes established between the peer MIS entities for protecting MIS messages at the MIS protocol layer. An MIS SA is established via Transport Layer Security (TLS) handshake, extensible authentication protocol (EAP) execution, or via a group key distribution mechanism using group key block (GKB) where all of the TLS handshake, EAP execution, and group key distribution take place over the MIS protocol. When an MIS SA is established via TLS handshake, the TLS master key and its child keys, TLS random values, and the TLS ciphersuite negotiated in the TLS handshake are a part of the MIS SA. When an MIS SA is established via EAP execution, a master session key (MSK) or re-authentication master

session key (rMSK) and its child keys, MIS random values, and the MIS ciphersuite negotiated between the peer MIS entities are associated with the MIS SA. When an MIS SA is established via group key distribution mechanism using GKB, the master group key and its child keys, and the MIS group ciphersuite indicated to the peer MIS entities are associated with the MIS SA.

MIS service access authentication server: An authentication server used to execute the media independent service (MIS) service access authentication. *See:* **authentication server**.

MIS service access authentication: An authentication process that authorizes the access to media independent services (MISs).

mobile node (MN): Communication node that is capable of changing its point of attachment from one link to another.

neighborhood network: The area of interest in which the network discovery and selection entity seeks to determine the available coverage of a wired/wireless network with identical or different link-layer technologies.

network entity: A communication node inside the network.

network point of attachment (network PoA, or **PoA):** The network-side endpoint of a layer 2 link that includes a mobile node as the other endpoint. *See also:* **candidate PoA**; **serving PoA**; **target PoA**.

network selection: The process by which a mobile node or a network entity makes a decision to connect to a specific network (possibly out of many available) based on a policy configured in the mobile node and/or obtained from the network.

network selector: The entity that undertakes the network selection decisions that can lead to a handover.

network-controlled handover: A handover where the network has the primary control over the handover process.

network-initiated handover: The network initiates the handover process by indicating to the mobile node that the handover is necessary or desired.

node index: A binary string assigned to a node in a group management tree.

node key: A cryptographic key assigned to a node in a group management tree. It is a device key and is shared by multiple devices when the node is not a leaf node.

operator identifier (operator ID): An identifier of the access or core network provider.

perfect binary tree: A binary tree with all leaf nodes at the same depth and all internal nodes having two children.

proactive authentication: A media-specific authentication with the candidate network(s) executed prior to a handover to one of the candidate networks.

protection mechanisms for media independent service (MIS) messages: A protection mechanism that is applied to MIS protocol data unit (PDU) using an encryption algorithm, an integrity protection algorithm, an authenticated encryption algorithm, or a combination of an encryption algorithm and an integrity protection algorithm.

protocol implementation conformance statement (PICS) proforma: A normative document to express in compact form the static conformance requirements of a specification. As such, it serves as a reference to the static conformance review.

seamless handover: A handover associated with a link switch between points of attachment, where the mobile node either experiences no degradation in service quality, security, and capabilities, or experiences some degradation in service parameters that is mutually acceptable to the mobile subscriber and to the network that serves the newly connected interface.

security association identifier (SAID): An identifier of an media independent service (MIS) security association. When an SA is established through Transport Layer Security (TLS), it is the TLS session ID. When an SA is generated through an extensible authentication protocol (EAP) execution, it is assigned by the authenticator and the ID value is an octet string unique for a pair of MIS functions. When an SA is generated via group key distribution mechanism using group key block (GKB), it is assigned by the group manager (GM) and the ID value is an octet string unique for groups of MIS functions.

serving point of attachment (serving PoA): The PoA of the current link being used by the mobile node.

serving PoS: An MIS PoS that is currently providing the MIS services to the mobile node.

soft handover: Handover where facilities for supporting traffic flows are continuously available while the mobile node link-layer connection transfers from the serving point of attachment to the target point of attachment. The network allocates transport facilities to the target point of attachment prior to the occurrence of the link switch event (make-before-break).

static conformance requirement: One of the requirements that specify the limitations on the combinations of implemented capabilities permitted in a real open system, which is claimed to conform to the relevant specification(s).

static conformance review: A review of the extent to which the static conformance requirements are claimed to be supported by the system under test, by comparing the answers in the implementation conformance statement(s) and the system conformance statement with the static conformance requirements expressed in the relevant specifications.

target point of attachment (target PoA): A candidate PoA that has been selected to become the new serving PoA.

two-member group: A group consisting of exactly two members.

uniform resource identifier (URI): A compact sequence of characters that identifies an abstract or physical resource including video.

vertical handovers: A handover where the mobile node moves between point of attachments of different link types, such as from universal mobile telecommunications system (UMTS) to wireless area network (WLAN). *Syn:* **inter-technology handovers**.

4. Abbreviations and acronyms

3G	3rd generation
3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AAA	authentication, authorization, and accounting
ACK	acknowledgement
AES	advanced encryption standard
AID	action identifier
ANDSF	access network discovery and selection function
ANQP	access network query protocol
AP	access point

AR	access router
AS	authentication server
ATSC-M/H	advanced television system committee—mobile handheld
BS	base station
CA	certificate authority
CBC	cipher block chaining
CCM	counter with cipher block chaining (CBC) message authentication code
CID	Company ID
CoA	care-of address
CoS	class of service
CS	convergence sublayer/command service
DCD	downlink channel descriptor
DHCP	dynamic host configuration protocol
DO	downlink only
DTLS	datagram transport layer security
DVB	digital video broadcasting
EAP	extensible authentication protocol
ERP	EAP re-authentication protocol
ES	event service
ESS	extended service set
FA	foreign agent
GKB	group key block
GM	group manager
GPRS	general packet radio service
GSM	global system for mobile communication
HESSID	homogenous extended service set ID
HMAC	keyed-hash message authentication code
IETF	Internet Engineering Task Force

IP	internet protocol
IPsec	internet protocol security
IS	information service
ITU	International Telecommunications Union
IV	initialization vector
L1	layer 1 (physical layer [PHY])
L2	layer 2 (medium access control [MAC] and/or logical link control [LLC])
LAN	local area network
LbyR	location by reference
LCP	location configuration protocol
LLC	logical link control
LSAP	logical link control service access point
LTE	Long-Term Evolution
MAC	medium access control
MGK	master group key
MIAK	media independent authentication key
MIB	management information base
MIC	message integrity code
MICS	media independent command services
MIEK	media independent encryption key
MIES	media independent event services
MIGKVK	media independent group key verification key
MIGSK	media independent group session key
MIIK	media independent integrity key
MIIS	media independent information service
MIP	mobile internet protocol
MIS	media independent service
MISF	media independent service function

MISK	media independent session key
MLME	medium access control (MAC) layer management entity
MN	mobile node
MPLS	multi-protocol label switching
MS	mobile station
MSA	media-specific authenticator
MSB	most significant bit
MSDU	medium access control (MAC) service data unit
MSGCF	medium access control (MAC) state generic convergence function
MSK	master session key
MSPMK	media-specific pairwise master key
MSRK	media-specific root key
N/A	not applicable
NAI	network access identifier
NAS	network access server
NCMS	network control and management system
OUI	organizationally unique identifier
PDU	protocol data unit
PHY	physical layer
PLME	physical layer management entity
PLMN	public land mobile network
PoA	point of attachment
PoS	point of service
PPP	point-to-point protocol
PRF	pseudorandom function
PSAP	public safety answering point
QoS	quality of service
RDF	resource description framework

RFC	request for comment
RLC	radio link control
rMSK	re-authentication master session key
RNC	radio network controller
RSNA	robust security network association
RSSI	received signal strength indication
SA	security association
SAID	security association identifier
SAP	service access point
SCTP	stream control transmission protocol
SDO	standards development organization
SDU	service data unit
SIB	system information block
SHA	secure hash algorithm
SID	service identifier
SINR	signal over interference plus noise ratio
SME	station management entity
SNR	signal-to-noise ratio
SRHO	single radio handover
STA	station
TCP	transmission control protocol
T-DMB	terrestrial-digital media broadcast
TLS	transport layer security
TLV	type length value (a form of encoding, or an item encoded using that coding)
UDP	user datagram protocol
UE	user equipment
UIR	unauthenticated information request
UMTS	universal mobile telecommunications system

URI	uniform resource identifier
URL	uniform resource locator
WLAN	wireless local area network
XML	extensible mark-up language

5. General architecture

5.1 Introduction

5.1.1 General

This standard supports a framework and necessary knobs to support several uses cases that are described in IEEE Std 802.21.1™-2017. In general, these use cases can be implemented and realized independently, but several additional functional entities need to be implemented in order to realize the overall system. While many of such functionalities do not fall within the scope of this standard, it is beneficial to understand certain functionalities so that the role and purpose of the media independent services (MISs) are clear and how they can be better utilized. The following subclauses give an overview of a few such functionalities.

5.1.2 Application class

Various applications have different tolerance characteristics for delay and data loss. Application-aware service decisions can be possible by making a provision for such characteristics. For example, when a network transition due to impending handover is made during the pause phase of a conversation in an active voice call, the perceptible interruption in the service is minimized.

5.1.3 Quality of service

The quality of the service (QoS) experienced by an application depends on the accuracy, speed, and availability of the information transfer in the communication channel. This standard provides support for fulfilling application QoS requirements for a variety of use cases. For example, QoS may be more relevant for handover use case than others as evident from the following description: there are two aspects of QoS to consider. First, there is the QoS experienced by an application during a handover. Second, there is the QoS considered as part of a handover decision. This standard includes mechanisms that support both aspects of QoS toward enabling seamless handover; however, the media independent service function (MISF) alone cannot guarantee seamless handover. Depending on the QoS requirements of the end-to-end application, seamless handover implies minimizing the handover latency and packet loss so as to minimize the end-to-end delay and the loss of transmitted information. Seamless mobility also implies the timely assessment of network conditions, such as the monitoring of packet loss on the current link and signal strength from both current and target networks, in order to optimize the handover decision and its execution.

5.1.4 Power management

This standard allows the mobile node (MN) to discover different types of wireless networks (e.g., IEEE 802.11, IEEE 802.16, and 3rd Generation Partnership Project [3GPP] networks), avoiding powering-up of multiple radios and/or excessive scanning at the radios. Thus, this standard minimizes power consumed by mobile devices in the discovery of potential candidate networks. Specific power management mechanisms deployed depend on individual link-layer technologies, and the potential power management benefits from this standard only extend to the discovery of wireless networks.

5.1.5 Handover policy

In the handover use case described in IEEE Std 802.21.1-2017, the primary role of the MISF is to facilitate handovers and provide intelligence to the network selector entity. The MISF aids the network selector entity with the help of the event service, command service, and information service. The network selector entity, and the handover policies that control handovers, are outside the scope of this standard.

5.1.6 Proactive authentication and key establishment

This standard provides mechanisms for a mobile node to conduct a proactive authentication and key establishment with the candidate network authenticator(s) and point(s) of attachment (PoA[s]). The proactive authentication is conducted through media-specific network access authentication where authentication messages are exchanged between authentication end-points via a point of service (PoS). The MIS protocol is used for encapsulating the authentication messages between the MN and the PoS. A successful proactive authentication and key establishment allow a PoA in the target network to obtain a key(s) to protect the communication link between the mobile node and the PoA after the handover, for example.

5.1.7 Group communication

There are use cases where a set of nodes need to be managed as a group via a multicast communication. For example, during configuration update (e.g., firmware update) control messages are sent to a group of nodes instead of a single node. In other scenarios, a set of nodes moves like a group between network points of attachment. Examples of this scenario are: networks of sensors/actuators that move between production and management networks, a set of nodes in a mesh network that moves as a group from one gateway node to another, a group of nodes that travels together in a transportation medium while changing the network point of attachment. To manage the configuration or the node movement in a bandwidth efficient manner when the network is performing failover, failback, configuration, and other management operations, multicast-based group communication is required.

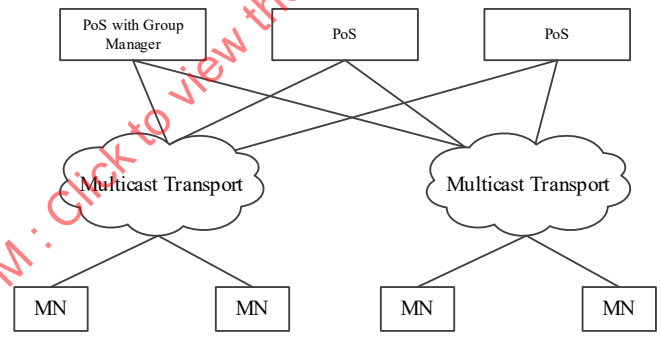


Figure 2—Group communication functional entities

This standard allows network nodes (e.g., PoS with group manager) to communicate media independent service commands to a group of MNs and PoSs via a multicast transport in a secure way. The standard defines primitives and corresponding messages for managing the multicast group membership (e.g., join, leave, and update the group membership) and provides mechanisms for managing the multicast group keys. Figure 2 shows the logical view of the functional entities that are involved in a multicast-based group communication. The group manager, as an MIS user in a PoS, is responsible for issuing the group manipulation commands in addition to generating and updating the master group keys and managing the group. PoSs (including the PoS with the group manager) and MNs send group addressed messages (see 8.3.2 and 9.6 for detailed rules on sending group addressed messages) via a multicast transport that is made available by the underlying network. This standard does not specify any transport protocol and the choice of transport protocol is left to the implementers.

5.2 General design principles

5.2.1 MISF design principles

This standard is based on the following general design principles:

a) MISF is a logical entity that facilitates decision-making processes for handover and other use cases. MIS users make use case specific decisions based on inputs from the MISF.

b) MISF provides abstracted services to higher layers. The service primitives defined by this interface are based on the technology-specific protocol entities of the different access networks. The MISF communicates with the lower layers of the protocol stack through technology-specific interfaces.

c) Higher layer applications (also known as MIS users) specify handover and other use case specific signaling mechanisms to MISF. The role of this standard is to serve as a framework to maximize the efficiency of other use cases by providing appropriate link-layer intelligence and network information as required by the higher layer applications.

d) The standard provides support for remote events and other message exchanges in a secure manner. Events are advisory in nature. For example, the decision whether to cause a handover or not based on these events is outside the scope of this standard.

e) The standard supports transparent operation with legacy equipment. IEEE 802.21–compatible equipment should be able to co-exist with legacy equipment.

5.2.2 QoS design principles

In the context of this standard it is assumed that applications communicate via a communication channel that is considered to be composed of several connected segments, each under a possibly different but cooperative administrative authority. Examples of such channels (e.g., for internet protocol [IP] traffic) have been detailed in ITU-T Recommendation Y.1540.

It is generally accepted that, based on the required accuracy of information transfer, applications are grouped into a small number of behavioral sets (ITU-T Recommendation Y.1540) called class of service (CoS). Support for differentiation via CoS is pervasive in many of the IEEE 802–based standards (IEEE Std 802.11-2012, IEEE Std 802.1Q-2014, IEEE Std 802.16-2012, etc.).

It is assumed that the classes of service definitions used within this standard conform to ITU-T Recommendation Y.1540.

5.3 MISF service overview

5.3.1 General

This standard defines services that comprise the MISF service; these services facilitate different use cases as described in IEEE Std 802.21.1-2017.

a) A media independent event service (MIES) that provides event classification, event filtering, and event reporting corresponding to dynamic changes in link characteristics, link status, and link quality.

b) A media independent command service (MICS) that enables MIS users to manage and control link behavior relevant to handovers and mobility.

c) A media independent information service (MIIS) that provides details on the characteristics and services provided by the serving and neighboring networks. The information enables effective system access and effective handover decisions.

The MISF provides asynchronous and synchronous services through well-defined SAPs for link layers and MIS users. In the case of a system with multiple network interfaces of arbitrary type, the MIS users use the event service, command service, and information service provided by MISF to manage, determine, and control the state of the underlying interfaces.

These services provided by MISF help the MIS users to satisfy relevant use cases described in IEEE Std 802.21.1-2017 that help maintaining service continuity, service adaptation to varying quality of service, securing and managing individual and group communication, battery life conservation, network discovery, and link selection. In a system containing heterogeneous network interfaces of IEEE 802 types and cellular (e.g., 3GPP, 3rd Generation Partnership Project 2 [3GPP2]) types, the MISF helps the MIS users to implement effective procedures to couple services across heterogeneous network interfaces. MIS users utilize services provided by the MISF across different entities to query resources required for a service operation between heterogeneous networks.

5.3.2 Media independent event service

5.3.2.1 General

Events indicate changes in state and transmission behavior of the physical, data link, and logical link layers, or predict state changes of these layers. The event service is also used to indicate management actions or command status on the part of the network or some management entity.

5.3.2.2 Event origination

Events originate from the MISF (MIS events) or any lower layer (Link events) within the protocol stack of an MN or network node, as shown in Figure 12 as described in 6.3.1.

5.3.2.3 Event destination

The destination of an event is the MISF or any upper layer entity. The recipient of the event is located within the node that originated the event or within a remote node. The destination of an event is established with a subscription mechanism that enables an MN or network node to subscribe its interest in particular event types.

5.3.2.4 Event service flow

In the case of local events, messages often propagate from the lower layers (e.g., PHY, MAC) to the MISF and from MISF to any upper layer. In case of remote events, messages propagate from the MISF in one protocol stack to the MISF in the peer protocol stack. One of the protocol stacks is present in an MN while the other is present in a fixed network entity. This network entity is the point of attachment or any node not directly connected to the other protocol stack.

5.3.2.5 Event service use cases and functions

The event service is used to detect the need for changing a link. For example, an indication that the link is unable to carry medium access control (MAC) service data units (SDUs) at some point in the near future is used by MIS users to prepare a new point of attachment ahead of the current point of attachment ceasing to carry frames. This has the potential to reduce the time needed for example for handover between attachment points. Events carry additional context data such as a layer 2 (L2) (MAC and/or logical link control [LLC]) identifier or L3 identifier. For example, a Link_Up event also carries a new IP address acquisition indication that informs the upper layers of the need to initiate a layer 3 handover.

5.3.3 Media independent command service

5.3.3.1 General

The command service enables higher layers to control the physical, data link, and logical link layers (also known as *lower layers*). The higher layers control the reconfiguration or selection of an appropriate link through a set of handover commands. When an MISF receives a command, it is always expected to execute the command.

5.3.3.2 Command origination

Commands are invoked by MIS users (MIS commands), as well as by the MISF itself (link commands), as shown in Figure 15.

5.3.3.3 Command destination

The destination of a command is the MISF or any lower layer. The recipient of a command is located within the protocol stack that originated the command, or within a remote protocol stack.

5.3.3.4 Command service flow

In the case of local commands, messages often propagate from the MIS users (e.g., policy engine) to the MISF and then from MISF to lower layers. In the case of remote commands, messages propagate from MIS users via MISF in one protocol stack to the MISF in a peer protocol stack (with the use of the MIS protocol). One of the protocol stacks is present in an MN while the other is present in a fixed network entity. This network entity is either a point of attachment or any node not directly connected to the other protocol stack.

5.3.3.5 Command service use cases and function

The commands generally carry the upper layer decisions to the lower layers on the local device entity or at the remote entity. For example, the command service is used by the policy engine of an entity in the network to request an MN to switch between links (remote command to lower layers on MN protocol stack).

For example, during network selection, the MN and the network need to exchange information about available candidate networks and select the best network. Another example: during group membership change, the MN and the network need to exchange information about group membership and group key so to allow it to join the appropriate group.

This standard supports a set of media independent commands that help with network selection under different conditions. These commands allow both the MN and the network to initiate services' specific uses and exchange information about available networks and negotiate the best available network under different conditions.

5.3.4 Media independent information service

The media independent information service (MIIS) provides a framework and corresponding mechanisms by which an MISF entity is able to discover and obtain necessary network information existing within a geographical area to facilitate the use cases.

The neighboring network information discovered and obtained by this framework and mechanisms is also used in conjunction with user and network operator policies for optimum initial network selection and access (attachment), or network re-selection in idle mode.

MIIS primarily provides a set of information elements (IEs), the information structure and its representation, and a query/response type of mechanism (pull mode) for information transfer. MIIS also supports a push mode wherein the information is pushed to the MN by the operator. The information can be present in an information server from where the MISF in the MN accesses it. The definition of the information server is outside the scope of this standard. In other cases, information can be present locally in the MN, and can be learned by the MN or pre-provisioned, or both. The definition of and indexing of such a local database, as well as the regime for maintaining it or accessing it, are outside the scope of this standard.

The information is made available via both lower and higher layers. Information is made available at L2 through a secure port.

In certain scenarios information is not accessed at L2, or the information available at L2 is not sufficient to make an intelligent decision. In such cases information can be accessed via higher layers. Hence this standard enables both L2 and L3 transport options for information access and provides message and data security, such as integrity and confidentiality.

MIIS typically provides static link-layer parameters such as channel information, the MAC address, and security information of a point of attachment (PoA). Information about available higher layer services in a network also helps in more effective decision and management such as handover, allocation of radio resources, and distribution of group keys.

The information provided by MIIS conforms to the structure and semantics specified within this standard. MIIS specifies a common (or media independent) way of representing this information across different technologies by using a standardized format such as extensible mark-up language (XML) or binary encoding. A structure of information is defined as a schema.

MIIS provides the ability to access information about all networks in a geographical area from any single L2 network, depending on how the IEEE 802.21 MIIS service is implemented. MIIS either relies on existing access media-specific transports and security mechanisms or L3 transport and L3 security mechanisms to provide access to the information. How this information is developed and deployed in a given network is outside the scope of the standard. Typically, in a heterogeneous network composed of multiple media types, the network selector or higher layer applications (e.g., mobility management entity) should collect information from different media types and assemble a consolidated view of the network.

Some networks, such as the cellular networks, already have an existing means of detecting a list of neighborhood network base stations within the vicinity of an area via the broadcast control channel. Some IEEE standards define similar means and support for MNs in detecting a list of neighborhood network access points within the vicinity of an area via either beaconing or via the broadcast of MAC management messages. MIIS defines a unified mechanism to the higher layer entities to provide candidate networks information in a heterogeneous network environment by a given geographical location. However, the algorithm for deciding what information to provide is out of scope.

5.4 Media independent service reference framework

5.4.1 General

The following subclause describes the key points with regards to communication between different MISF entities in the MN and the network. The reference points in this subclause (5.4) are for illustration only. This subclause does not define any specific deployed network system architecture.

5.4.2 MISF communication model

MIS functions communicate with each other for various purposes. The MN exchanges MIS information with its MIS point of service (PoS). The MISF in any network entity becomes an MIS PoS when it communicates directly with an MN-based MISF. When an MISF in a network entity does not have a direct

connection to the MN, it does not act as an MIS PoS for that particular MN. However, the same MIS network entity is capable of acting as MIS PoS for a different MN.

When an MN has multiple L2 interfaces, MISF communication need not take place on all L2 interfaces of an MIS-capable MN. The MN uses L2 transport for exchanging MIS information with an MIS PoS that resides in the same network entity as its Network PoA. The MN uses L3 transport for exchanging MIS information with an MIS PoS that does not reside in the same network entity as its Network PoA. The framework supports use of either L2 or L3 mechanisms for communication among MIS network entities.

Figure 3 shows the MISF communication model. The model shows MISFs in different roles and the communication relationships among them. The communication relationship shown in Figure 3 applies only to MISFs. It is important to note that each of the communication relationships in the communication model does not imply a particular transport mechanism. Rather, a communication relationship only intends to show that passing MISF-related information is possible between the two different MISFs. Moreover, each communication relationship shown in the diagram encompasses different types of interfaces, different transport mechanisms used (e.g., L2, L3), and different MISF service related content being passed (e.g., MIIS, MICS, or MIES).

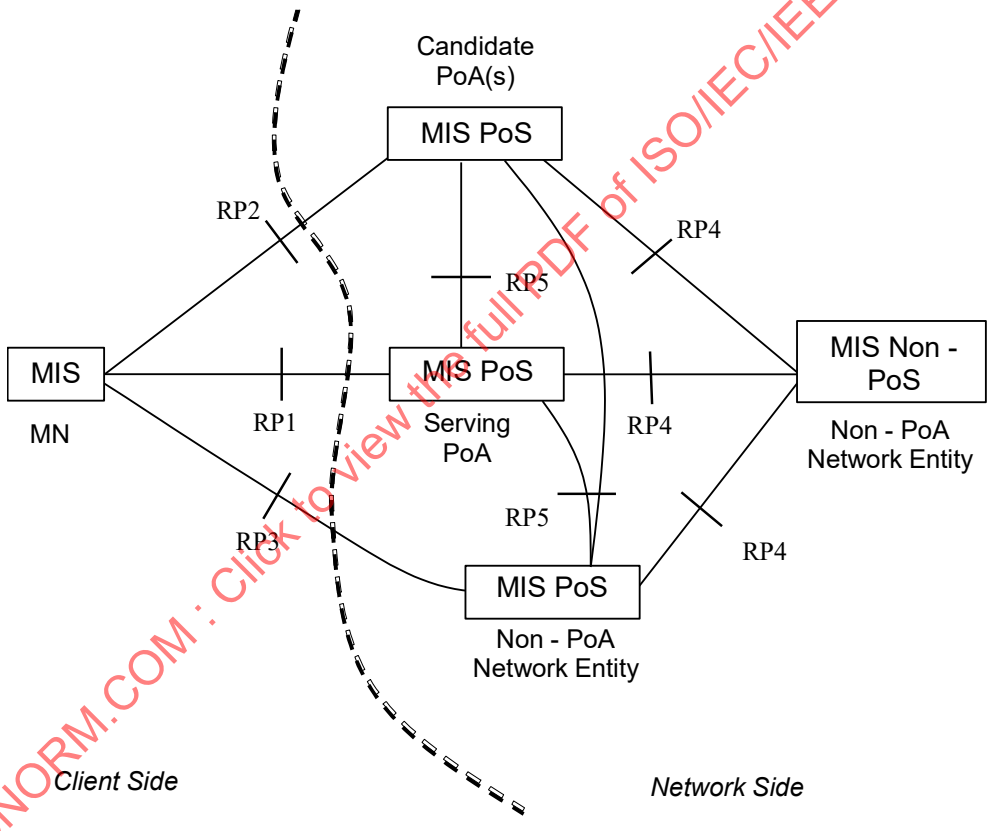


Figure 3—MISF communication model

The communication model assigns different roles to the MISF depending on its position in the system.

a) MISF on the MN
b) MIS PoS on the network entity that includes the serving PoA of the MN
c) MIS PoS on the network entity that includes a candidate PoA for the MN
d) MIS PoS on a network entity that does not include a PoA for the MN
e) MIS non-PoS on a network entity that does not include a PoA for the MN

The communication model also identifies the following reference points between different instances of MISFs (see Table 1).

— **Reference point RP1:** Reference point RP1 refers to MISF procedures between the MISF on the MN and the MIS PoS on the network entity of its serving PoA. RP1 encompasses communication interfaces over both L2 and L3 and above. MISF content passed over RP1 are related to MIIS, MIES, or MICS.

— **Reference point RP2:** Reference point RP2 refers to MISF procedures between the MISF on the MN and the MIS PoS on the network entity of a candidate PoA. RP2 encompasses communication interfaces over both L2 and L3 and above. MISF content passed over RP2 are related to MIIS, MIES, or MICS.

— **Reference point RP3:** Reference point RP3 refers to MISF procedures between the MISF on the MN and the MIS PoS on a non-PoA network entity. RP3 encompasses communication interfaces over L3 and above and possibly L2 transport protocols like Ethernet bridging, or multi-protocol label switching (MPLS). MISF content passed over RP3 are related to MIIS, MIES, or MICS.

— **Reference point RP4:** Reference point RP4 refers to MISF procedures between an MIS PoS in a Network Entity and an MIS non-PoS instance in another network entity. RP4 encompasses communication interfaces over L3 and above. MISF content passed over RP4 are related to MIIS, MIES, or MICS.

— **Reference point RP5:** Reference point RP5 refers to MISF procedures between two MIS PoS instances in different Network Entities. RP5 encompasses communication interfaces over L3 and above. MISF content passed over RP5 are related to MIIS, MIES, or MICS.

Table 1—Summary of reference points

Reference point	Description
RP1	Between the MISF on an MN and an MIS PoS on the network entity of the serving PoA
RP2	Between the MISF on an MN and an MIS PoS on the network entity of the candidate PoA
RP3	Between the MISF on an MN and an MIS PoS on a non-PoA network entity
RP4	Between an MIS PoS and an MIS non-PoS instance in different network entities
RP5	Between two MIS PoS instances in different network entities

All references point definitions are within the scope of this standard. Annex C provides a mapping of various MIS messages to the reference points.

5.5 MISF reference models for link-layer technologies

The MISF provides asynchronous and synchronous services through well-defined service access points for MIS users. The following subclauses (5.5.1 through 5.5.7) describe the reference models for various link-layer technologies with MIS functionality.

5.5.1 IEEE 802 architectural considerations

The MIS reference models for different IEEE 802 technologies and the general MIS framework is designed to be consistent with the IEEE 802 Architecture for different link-layer technologies. The MIS function is a management entity that obtains link-layer information from lower layers of different protocol stacks and also from other remote nodes. The MIS function coordinates handover decision making with other peer MIS functions in the network.

The MIS protocol provides the capability for transferring MIS messages between peer MIS function entities at L2 or at L3. These messages transfer information about different available networks and also provide network switching, handover, and other use case specific IEEE Std 802.21.1-2017 capabilities across different networks. The MIS protocol encompasses IEEE 802 technologies such as IEEE 802.3, IEEE 802.11, and IEEE 802.16, and also other non-IEEE 802 technologies such as those specified by 3GPP and 3GPP2 standards. In this sense, the MIS protocol has different scope and functionality than the Link Layer Discovery Protocol (LLDP) as specified by IEEE Std 802.1AB-2009.

5.5.2 General MISF reference model and SAPs

Figure 4 illustrates the position of the MISF in a protocol stack and the interaction of the MISF with other elements of the system. All exchanges between the MISF and other functional entities occur through service primitives, grouped in service access points (SAPs).

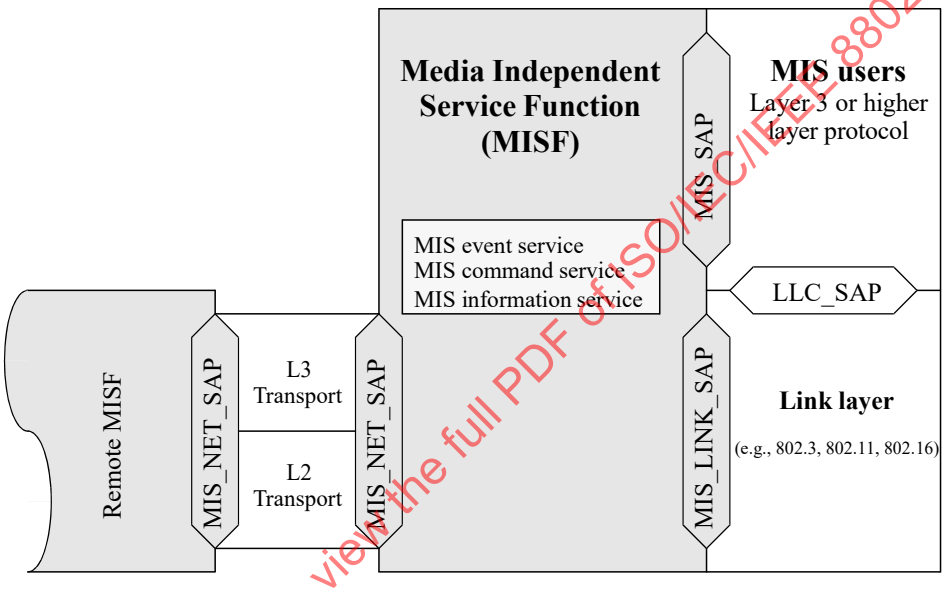


Figure 4—General MISF reference model and SAPs

The media agnostic general MIS reference model includes the following SAPs:

a) MIS_SAP: Media independent interface of MISF with the upper layers of the protocol stack.

b) MIS_LINK_SAP: Abstract media dependent interface of MISF with the lower layers of the media-specific protocol stacks.

c) MIS_NET_SAP: Abstract media dependent interface of MISF that provides transport services over the data plane on the local node, supporting the exchange of MIS information and messages with the remote MISF. For all transport services over L2, the MIS_NET_SAP uses the primitives specified by the MIS_LINK_SAP.

In the media-specific reference models, the media independent SAP (MIS_SAP) always maintains the same name and same set of primitives. The media dependent SAP (which is a technology-specific instantiation of the MIS_LINK_SAP), assumes media-specific names and sets of primitives, often reusing names and primitives that already exist in the respective media-specific existing lower layer SAPs. Primitives defined in MIS_LINK_SAP result in amendments to media-specific SAPs due to additional functionality being defined for interfacing with the MISF. All communications of the MISF with the lower layers of media-specific protocol stacks take place through media-specific instantiations of MIS_LINK_SAP.

The message exchanges between peer MISF instances, in particular the type of transport that they use, are sensitive to several factors, such as the nature of the network nodes that contain the peer MISF instances (whether or not one of the two is an MN or a PoA), the nature of the access network (whether IEEE 802 or 3G cellular), and the availability of MIS capabilities at the PoA.

Figure 5 presents a summary of the types of relationships that exists between the MISF and other functional components in the same network node.

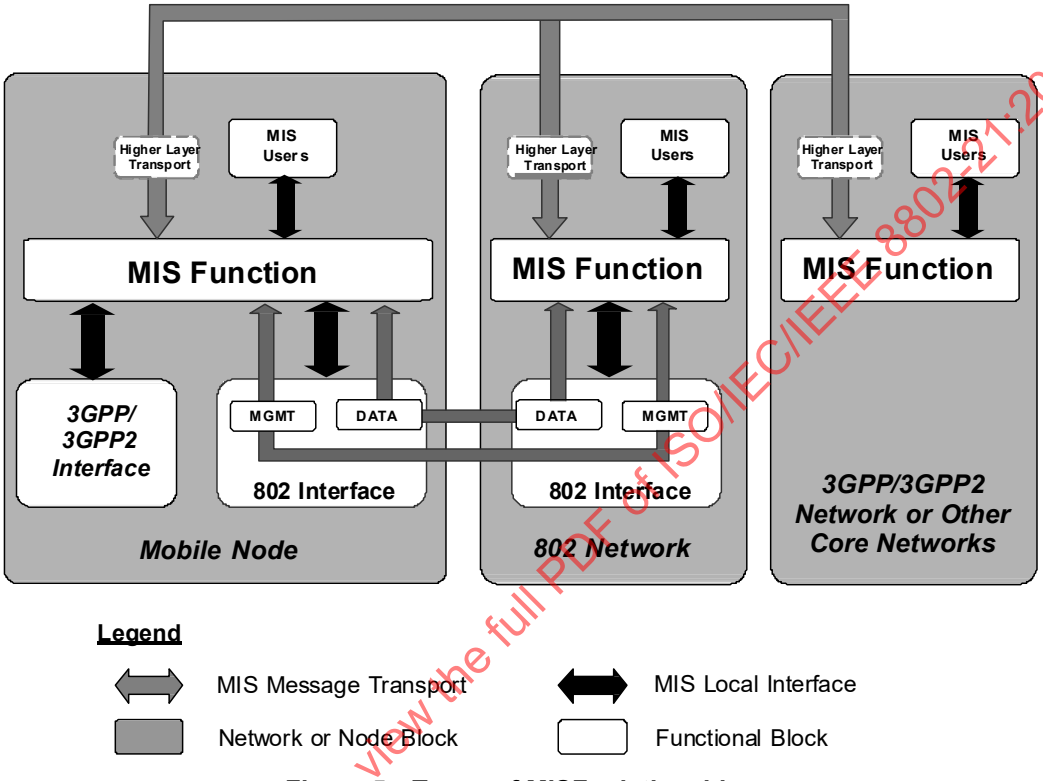


Figure 5—Types of MISF relationships

The general MIS reference model in Figure 4 enables a simple representation of the broad variety of MISF relationships shown in Figure 5. In the model, a mobility-management protocol stack is logically identified within each network node that includes an MISF instance. The provided abstraction makes it easy to isolate and represent the MIS relationships with all pre-existing functional entities within the same network node. Such relationships are both internal (with functional entities that, just like the MISF, share the logical inclusion in the mobility-management protocol) and external (with functional entities that belong to other planes).

Figure 5 shows how an MIS-enabled MN communicates with an MIS-enabled network. The gray arrows show the MIS signaling over the network, whereas the black arrows show local interactions between the MISF and lower and higher layers in the same network or node block. For a more detailed view of local interactions, please refer to technology-specific reference models and service access point in 5.5.3 through 5.5.7.

When connected to an IEEE 802 network, an MN directly uses L2 for exchanging MIS signaling, as the peer MISF can be embedded in a PoA. The MN does this for certain IEEE 802 networks even before being authenticated with the network. However, the MN can also use L3 for exchanging MIS signaling, for example in cases where the peer MISF is not located in the PoA, but deeper in the network.

When connected to a 3GPP or 3GPP2 network, an MN uses L3 transport to conduct MIS signaling.

5.5.3 MISF reference model for IEEE 802.3

The MISF reference model for IEEE 802.3 is illustrated in Figure 6. The transport of MISF services is supported over the data plane by use of existing primitives defined by the logical link control service access point (LSAP). There are no amendments specified in IEEE Std 802.3-2012 to support any link services defined over the MIS_LINK_SAP in this specification.

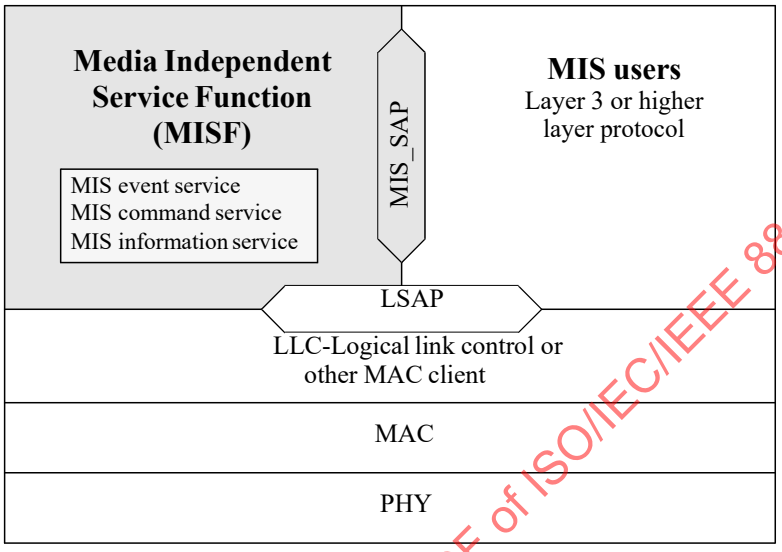


Figure 6—MIS reference model for IEEE 802.3

5.5.4 MISF reference model for IEEE 802.11

Figure 7 shows the MISF reference model for IEEE 802.11. The payload of MISF services over IEEE 802.11 is carried either in the data frames by using existing primitives defined by the LSAP or by using primitives defined by the MAC state generic convergence function (MSGCF) service access point (SAP) (MSGCF_SAP). The MSGCF has access to all management primitives and provides services to higher layers.

It should be noted that sending MISF payload over the LSAP is allowed only after successful authentication and association of the station to the access point (AP). Moreover, before the station has authenticated and associated with the AP, only MIS Information Service and MIS capability discovery messages are transported over the MSGCF_SAP.

The MIS_SAP specifies the interface of the MISF with MIS users.

5.5.5 MISF reference model for IEEE 802.16

Figure 8 shows the MISF for IEEE 802.16-based systems. The Management SAP (M_SAP) and Control SAP (C_SAP) are common between the MISF and network control and management system (NCMS).

The M_SAP specifies the interface between the MISF and the management plane and allows MISF payload to be encapsulated in management messages (such as MOB_MIS-MSG defined in IEEE Std 802.16-2012). The primitives specified by M_SAP are used by an MN to transfer packets to a base station (BS), both before and after it has completed the network entry procedures. The C_SAP specifies the interface between the MISF and control plane. M_SAP and C_SAP also transport MIS messages to peer MISF entities. The Convergence Sublayer SAP (CS_SAP) is used to transfer packets from higher layer protocol entities after appropriate connections have been established with the network.

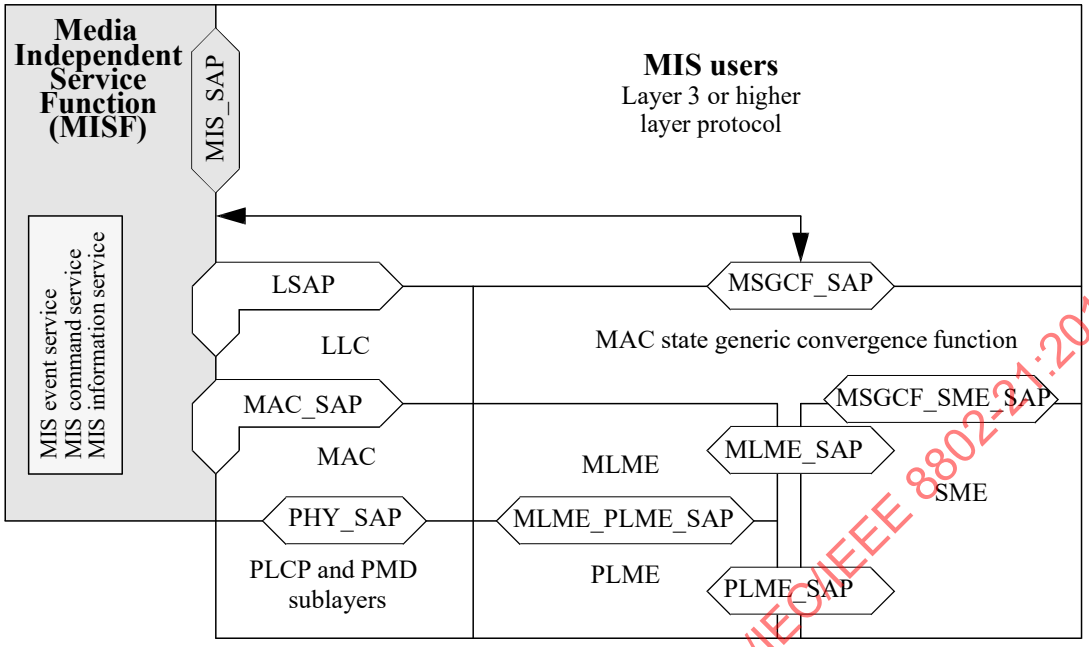


Figure 7—MIS reference model for IEEE 802.11

The MIS_SAP specifies the interface of the MISF with other higher layer entities such as transport layer, handover policy engine, and layer 3 mobility protocol.

In this model, C_SAP and M_SAP provide link services defined by MIS_LINK_SAP, C_SAP provides services before network entry, while CS_SAP provides services over the data plane after network entry.

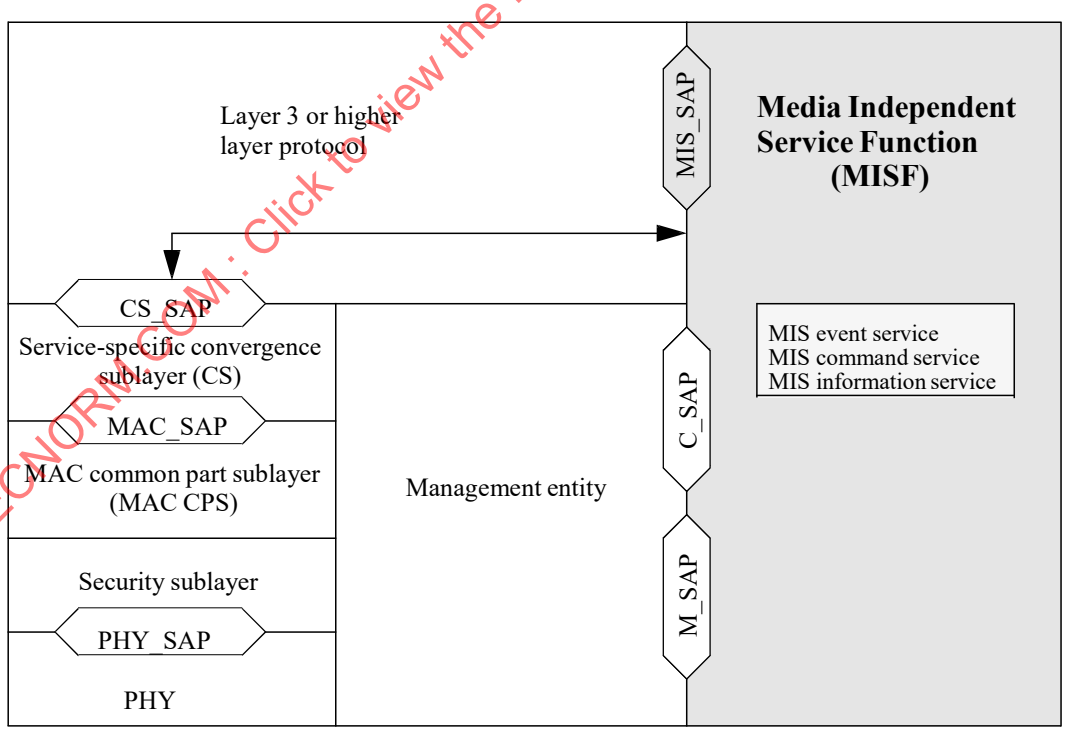


Figure 8—MIS reference model for IEEE 802.16

5.5.6 MISF reference model for 3GPP

Figure 9 illustrates the interaction between the MISF and the 3GPP-based systems. The MISF services are specified by the MIS_3GLINK_SAP. However, no new primitives or protocols need to be defined in the 3GPP specification for accessing these services. The MISF services are mapped to existing 3GPP signaling functions (see Table D.3). The architectural placement of the MISF is left to the 3GPP standard. Figure 9 is for illustrative purposes only and should not constrain implementations.

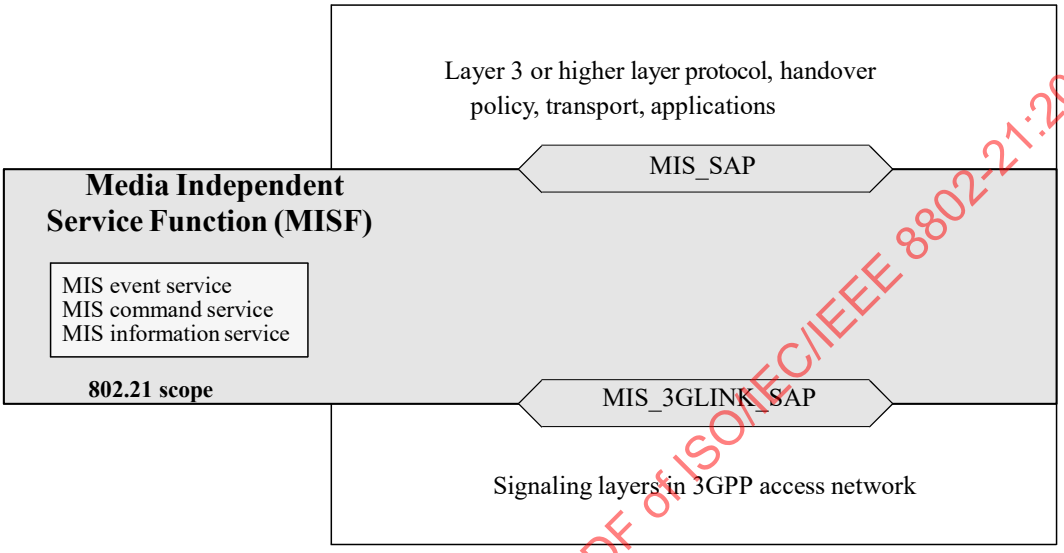


Figure 9—MIS reference model for 3GPP systems

5.5.7 MISF reference model for 3GPP2

Figure 10 illustrates the interaction between IEEE 802.21 services and 3GPP2-based systems. IEEE 802.21 services are accessed through the MIS_3GLINK_SAP. However, note that no new primitives or protocols need to be defined within the 3GPP2 specification. Instead, a mapping between IEEE 802.21 link-layer primitives and 3GPP2 primitives as defined in IETF RFC 1661 and 3GPP2 C.S0004-D is already established. Primitive information available from upper layer signaling and point-to-point protocol (PPP) are directly used by mapping LAC SAP and PPP SAP primitives to IEEE 802.21 service primitives in order to generate an event.

This mapping is illustrated in Table D.3, which provides an example of how 3GPP and 3GPP2 primitives are mapped to IEEE 802.21 primitives. For example, events received from the upper layer signaling through the LAC layer SAP such as “L2.Condition.Notification” are mapped and generated through the MIS_3GLINK_SAP as a Link_Up, Link_Down, or Link_Going_Down. Likewise, events generated at the PPP_SAP within the PPP layer, such as LCP-Link-Up or IPCP_LINK_OPEN, could be mapped and generated through the MIS_3GLINK_SAP as a Link_Up event.

It is noteworthy that there is no direct communication between the 3GPP2 physical layer (PHY) and MAC layers with the MISF. The architectural placement of any MISF is left to 3GPP2. Figure 10 is for illustrative purposes only and should not constrain implementations.

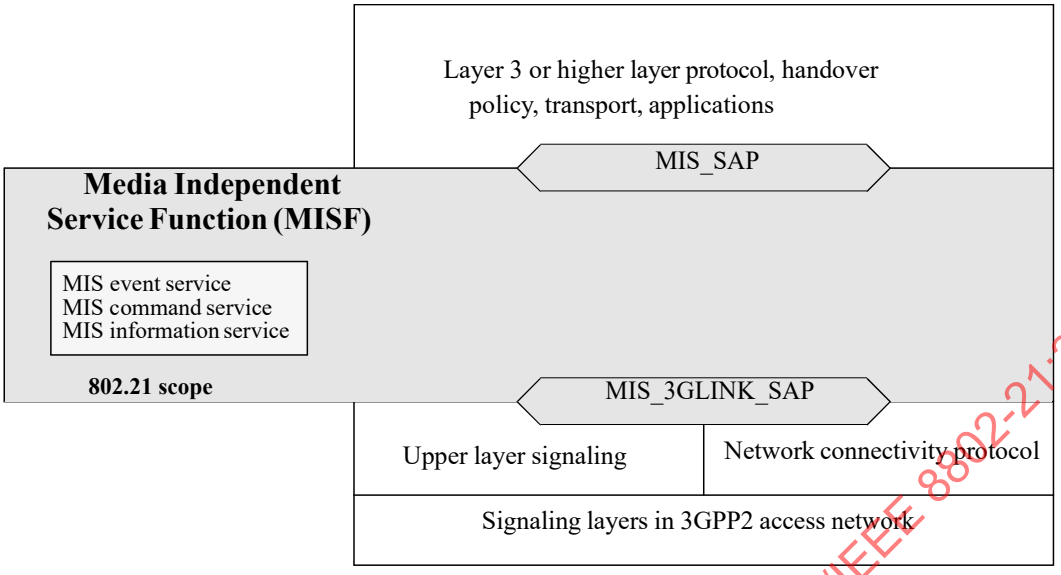


Figure 10—MIS reference model for 3GPP2 systems

5.6 Service access points (SAPs)

5.6.1 General

The MISF interfaces with other layers and functional planes using service access points (SAPs). Each SAP consists of a set of service primitives that specify the interactions between the service user and provider.

The specification of the MISF includes the definition of SAPs that are media independent and recommendations to define or extend other SAPs that are media dependent. Media independent SAPs (MIS-SAP) allow the MISF to provide services to the upper layers of the protocol stack, the network management plane, and the data bearer plane. The MIS_SAP and associated primitives provide the interface from MISF to the upper layers of the protocol stack. Upper layers need to subscribe with the MISF as users to receive MISF-generated events and also for link-layer events that originate at layers below the MISF but are passed on to MIS users through the MISF. MIS users directly send commands to the local MISF using the service primitives of the MIS_SAP. Communication between two MISFs relies on MIS protocol messages.

Media dependent SAPs allow the MISF to use services from the lower layers of the protocol stack and their management planes. All inputs (including the events) from the lower layers of the protocol stack into the MISF are provided through existing media-specific SAPs such as MAC SAPs, PHY SAPs, and LSAPs. Link commands generated by the MISF to control the PHY and MAC layers during the handover are part of the media-specific MAC/PHY SAPs and are already defined elsewhere.

Figure 11 shows the key MISF-related SAPs for different networks, which are as follows:

a) The MIS_SAP specifies a media independent interface between the MISF and upper layers of the protocol stack. The upper layers need to subscribe with the MISF as users to receive MISF-generated events and also for link-layer events that originate at layers below the MISF but are passed on to MISF users through the MISF. MISF users directly send commands to the local MISF using the service primitives of the MIS_SAP.

b) The MIS_LINK_SAP specifies an abstract media dependent interface between the MISF and lower layers media-specific protocol stacks of technologies such as IEEE 802.3, IEEE 802.11, IEEE 802.16, 3GPP, and 3GPP2. For different link-layer technologies, media-specific SAPs provide the functionality of MIS_LINK_SAP.

c) The MIS_NET_SAP specifies an abstract media dependent interface of the MISF that provides transport services over the data plane on the local node, supporting the exchange of MIS information and messages with remote MISFs.

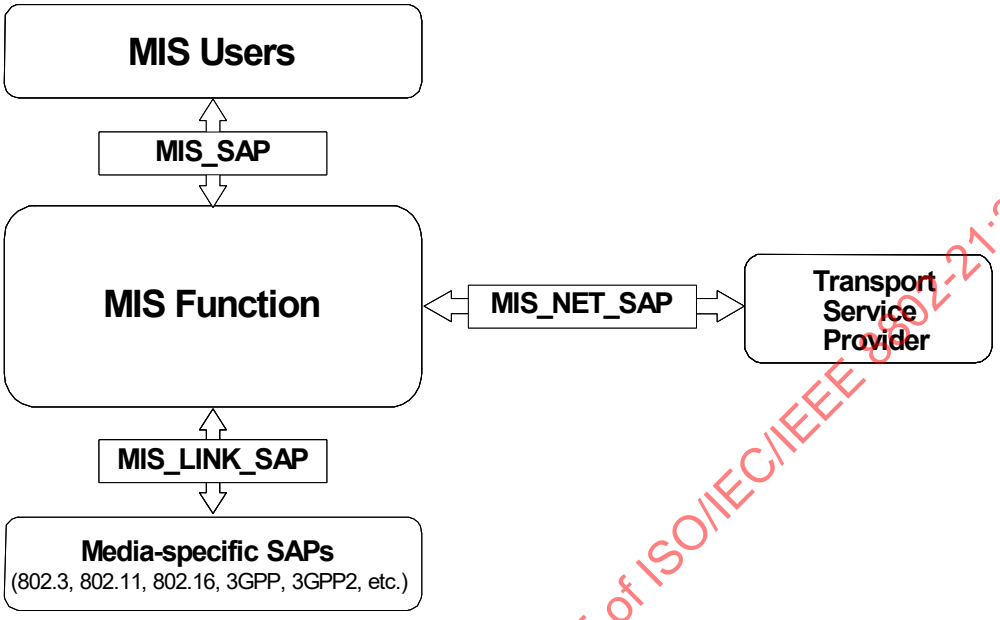


Figure 11—Relationship between different MISF SAPs

5.6.2 Media dependent SAPs

5.6.2.1 General

Each link-layer technology specifies its own technology-dependent SAPs. For each link-layer technology, the MIS_LINK_SAP maps to the technology-specific SAPs.

5.6.2.2 MIS_LINK_SAP

This SAP defines the abstract media dependent interface between MISF and different link-layer technologies. Amendments are suggested for different layer technology-specific SAPs based on the definition of this particular SAP.

5.6.2.3 MIS_NET_SAP

MIS_NET_SAP defines the abstract media dependent interface of the MISF that provides transport services over the data plane on the local node, supporting the exchange of MIS information and messages with remote MISFs. For L2, this SAP uses the primitives provided by MIS_LINK_SAP.

5.6.2.4 MLME_SAP

This SAP defines the interface between the MISF and the management plane of an IEEE 802.11 network. This SAP is used for sending MIS messages between the MISF and local link-layer entities, as well as between peer MISF entities.

5.6.2.5 C_SAP

The C_SAP, defined in IEEE Std 802.16-2012, provides the interface between the MISF and the IEEE 802.16 control plane. This SAP is used for MIS exchanges between the MISF and the lower layers of the management plane (as part of the IEEE 802.16 instantiation of the MIS_LINK_SAP).

5.6.2.6 M_SAP

The M_SAP, defined in IEEE Std 802.16-2012, provides the interface between the MISF and the IEEE 802.16 management plane functions.

5.6.2.7 MSGCF_SAP

This SAP provides services to MISF based on the IEEE 802.11 MAC state machines and interactions between the IEEE 802.11 sublayers.

5.6.2.8 MIS_3GLINK_SAP

This SAP works as an umbrella that defines the interface between the MISF and the different protocol elements of the cellular systems. The existing service primitives or media-specific SAPs as defined in 3GPP and 3GPP2 specifications are directly mapped to MISF services, and hence no new primitives need to be defined in these specifications. Table D.3 lists this mapping.

5.6.2.9 LSAP

The LSAP, defined in ISO/IEC 8802-2:1998, provides the interface between the MISF and the LLC sublayer in IEEE 802.3 and IEEE 802.11 networks. This SAP is used for local MIS exchanges between the MISF and the lower layers and for the L2 transport of MIS messages across IEEE 802 access links.

5.6.2.10 CS_SAP

The CS_SAP, defined in IEEE Std 802.16-2012, provides the interface between the MISF and the service CS (Convergence Sublayer) in IEEE 802.16 networks. This SAP is used for the L2 transport of MIS messages across IEEE 802.16 access links.

5.6.3 Media independent SAP: MIS_SAP

The MIS_SAP defines the media independent interface between the MISF and MIS users such as an upper layer mobility protocol or a group manager that might reside at higher layers or a higher layer transport entity as well. The definition of the MIS_SAP is required to define the scope and functionality of the MISF.

5.7 MIS protocol

5.7.1 General

MIS protocol defines the format of messages (i.e., MISF packet with header and payload) that are exchanged between remote MISF entities and the media independent mechanisms that support the delivery of these messages.

5.7.2 Ethertype use and encoding

All MIS protocol data units (PDUs) shall be identified using the MIS protocol Ethertype specified in Table 2.

Table 2—MIS protocol Ethernet type

Assignment	Value[a]
MIS protocol Ethernet type	8917

[a]This Ethertype value is expressed using the hexadecimal representation defined in IEEE Std 802.

5.7.3 Transport considerations

MIS protocol messages are sent over the data plane by use of a suitable transport mechanism at both layer 2 and layer 3. Layer 3 transport is supported using transmission control protocol (TCP)/user datagram protocol (UDP)/stream control transmission protocol (SCTP) protocols over IP. Layer 2 transport is supported with the Ethertype value set to that for MIS protocol. The data plane is available for transport after the MN has authenticated with the access network. In case of IEEE 802.11 and IEEE 802.16 networks, MIS protocol messages can also be sent before authentication over the management plane by using respective media-specific MAC management frames.

5.7.4 The generic MAC service with IEEE 802.1X

5.7.4.1 General

The generic MAC service in both IEEE 802.3 network and IEEE 802.11 robust security network association (RSNA) networks (which use IEEE Std 802.1X-2010 port-based network access control), goes through the controlled port after authentication and association. The uncontrolled port is in open access mode to allow only exchange of messages to perform authentication and secure connection association, whereas the controlled port is blocked until authentication and association are successful.

The MIS messages that pass through the LSAP are distinguished from other protocols with an Ethertype value of MIS protocol Ethernet type in the LLC header.

When the MIS protocol is used across IEEE 802.11 networks, MIS frames are exchanged when the STA has successfully authenticated and associated to the IEEE 802.11 access point.

5.7.4.2 Controlled port unblocked state: LSAP transport

After successful authentication and association, the controlled port is unblocked to the transport of authenticated messages. The MIS messages are then encapsulated into LLC protocol with Ethertype value of MIS protocol to pass through the controlled port. When LSAP receives MAC frames from the LLC layer, it checks the Ethertype of each frame to determine whether to send the frame to MISF protocol or to other protocols.

5.7.4.3 Controlled port blocked state

Until authentication has been completed, the controlled port is in blocked state so that MIS messages are not able to go through. However, in IEEE 802.11 and IEEE 802.16 networks transport of MIS messages (MIS_message types are limited to Information Service Query request/response, Event Service, and Command Service capability discovery only) are possible via the management plane prior to authentication.

6. MISF services

6.1 General

The MISF provides the media independent event service, the media independent command service, and the media independent information service that facilitate the use cases that are described in IEEE Std 802.21.1-2017. Clause 6 provides a general description of these services. These services are managed and configured through service management primitives, as discussed in 6.2. The corresponding configuration and management parameters are defined in an MIB, see Annex I.

6.2 Service management

6.2.1 General

Prior to providing the MIS services from one MISF to another, the MIS entities need to be configured properly. This is done through the following service management functions:

— MIS capability discovery
— MIS registration
— MIS service access authentication
— MIS event subscription
— MIS group configuration, manipulation, and key distribution

In order to know the services that are supported by an MIS peer, the MIS node performs MIS capability discovery. The MIS node performs MIS capability discovery with different MIS peers in order to decide which one to register with.

6.2.2 Service management primitives

Table 3 defines the set of service management primitives. A primitive is marked as local only (L), remote only (R), or local and remote (L, R), indicating whether it is invoked by a local MIS user, a remote MIS user, or both, respectively.

Table 3—Service management primitives

Service management primitive	(L)ocal, (R)emote	Defined in	Comments
MIS_Capability_Discover	L, R	7.4.1	Discover the capabilities of a local or remote MISF.
MIS_Register	R	7.4.2	Register with a remote MISF.
MIS_DeRegister	R	7.4.3	Deregister from a remote MISF.
MIS_Event_Subscribe	L, R	7.4.4	Subscribe for one or more MIS events with a local or remote MISF.
MIS_Event_Unsubscribe	L, R	7.4.5	Unsubscribe for one or more MIS events from a local or remote MISF.
MIS_Push_Key	L, R	7.4.17	Install a key in a remote PoA.
MIS_LL_Auth	L, R	7.4.18	Carry out a proactive authentication over MIS messages between the MN and the PoS using link-layer frames
MIS_Configuration_Update	R	7.4.19	This command is sent by a PoS to a group of MNs or other PoS(s) to update their configuration.
MIS_Pull_Group_Manipulate	R	7.4.20	This command is sent by an MN or a PoS to another PoS to create, delete, or update the group membership.
MIS_Push_Group_Manipulate	R	7.4.21	This command is sent by a PoS to a group of MNs or other PoS(s) to create, delete, or update a group membership.
MIS_Pull_Certificate	R	7.4.22	This command is generated by an MN or a PoS and it is used to request the sending of a certificate from the PoS to a destination PoS or MN.
MIS_Push_Certificate	R	7.4.23	This command is sent by a PoS to another PoS or an MN and it is used for sending of a certificate.
MIS_Revoke_Certificate	R	7.4.24	This command is sent by a PoS to a group of PoS(s) and/or an MN to revoke a certificate previously issued by the PoS.

6.2.3 MIS capability discovery

The MIS capability discovery procedure is used by an MIS user to discover a local or remote MISF's capabilities in terms of MIS services (Event Service, Command Service, and Information Service). MIS capability discovery is performed either through the MIS protocol or through media-specific mechanisms (i.e., IEEE 802.11 Beacon frames, IEEE 802.16 downlink channel descriptor [DCD], IEEE 802.11 management frames, or IEEE 802.16 management messages).

6.2.4 MIS registration

MIS registration is defined as a means of requesting access to specific MIS services. For example, in a network-controlled inter-technology handover framework, MIS registration can be used by an MN to declare its presence to a selected MIS PoS. MIS registration is mandatory for use with the MIS Command Service and the push mode of the MIS Information Service.

6.2.5 MIS event subscription

The MIS event subscription mechanism allows an MIS user to subscribe for a particular set of events that originates from a local or remote MISF. See 6.3.2 for a more detailed description of MIS event subscription.

6.2.6 Network communication

The network communication functions provide transport services over the data plane on the local node, supporting the exchange of MIS information and messages between the local and remote MISF. For

transport services over L2, MIS_NET_SAP utilizes the primitives specified by the MIS_LINK_SAP. For transport services over L3, the primitives are specified by MIS_NET_SAP. Please refer to 7.5 for more details on MIS_NET_SAP.

6.2.7 MIS group configuration and manipulation

The MIS group configuration and manipulation mechanisms enable a PoS to manage groups of MNs and/or PoSs in a secure way. These MNs and PoSs are reachable through a multicast transport. The primitives used to manage the membership to the groups and their security properties are called group manipulation commands throughout this specification. The group manipulation commands include the functionalities required to manage the group membership (join, leave, update operations) and to install the appropriate credentials to MNs and PoSs belonging to a particular group. Details on which MISF commands used for multicast group communication are described in 8.3.2.

6.3 Media independent event service

6.3.1 Introduction

In general, events relevant to handovers and other use cases originate from MAC, PHY, or MISF at the MN, at the network PoA, or at the PoS. Thus, the source of these events is either local or remote entity. A transport protocol is needed for supporting remote events. Security is another important consideration in such transport protocols.

Multiple higher layer entities can be interested in these events at the same time. Thus, these events can have multiple destinations. Higher layer entities can subscribe to receive event notifications from a particular event source. The MISF can help in dispatching these events to multiple destinations.

These events are treated as discrete events. As such there is no general event state machine. Event notifications are generated asynchronously. Thus, all MIS users and MISFs that want to receive event notifications need to subscribe to particular events.

From the recipient's perspective, these events are mostly "advisory" in nature and not "mandatory." The recipient is not obligated to act on these events. Layer 3 and above entities need to deal with reliability and robustness issues associated with these events. These events are likely to be used for horizontal handovers.

The Event Service is broadly divided into two categories, Link events and MIS events. Both Link and MIS events traverse from a lower to a higher layer. Link events are defined as events that originate from event source entities below the MISF and terminate at the MISF. Entities generating Link events include, but are not limited to, various IEEE 802-defined, 3GPP-defined, and 3GPP2-defined interfaces. Within the MISF, Link events propagate further, with or without additional processing, to MIS users that have subscribed for the specific events. MIS events are defined as events that originate from within the MISF, or they are Link events that are propagated by the MISF to the MIS users. This relationship is shown in Figure 12.

An event is either local or remote; a local event is one that propagates across different layers within the local protocol stack of an MIS entity, while a remote event is one that traverses across the network medium from one MIS entity to another MIS entity.

All Link events are local in nature and propagate from the local lower layer to the local MISF. MIS events are local or remote. A remote MIS event traverses the medium from a remote MISF to the local MISF and is then dispatched to local MIS users that have subscribed to this remote event, as shown in Figure 13.

A Link event that is received by the MISF can also be sent to a remote MIS entity as a remote MIS event.

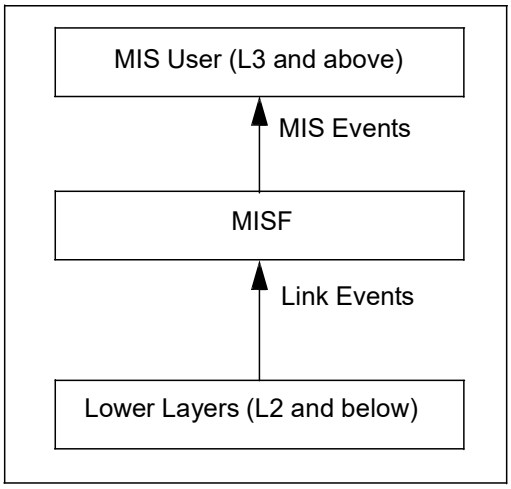


Figure 12—Link events and MIS events

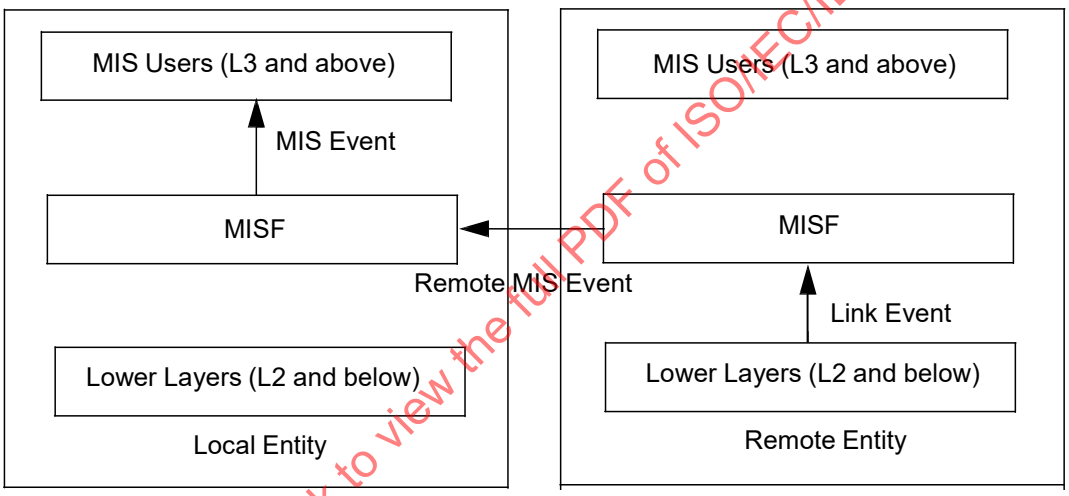


Figure 13—Remote MIS events

6.3.2 Event subscription

6.3.2.1 General

Event subscription provides a mechanism for upper layer entities to selectively receive events. Event subscription is divided into Link events subscription and MIS events subscription. Link events subscription is performed by the MISF with the event source entities in order to determine the events that each event source (link) is able to provide. MIS events subscription is performed by upper layer entities with the MISF to select the events to receive. It is possible for upper layer entities to subscribe for all existing events or notifications that are provided by the event source entity even if no additional processing of the event is done by the MISF.

6.3.2.2 Link events subscription

During initialization the MISF actively searches for pre-existing interfaces, devices, and modules that serve as Link event sources in the event service. In addition to the Link event source entities that are present during the bootstrapping stage, allowances are made for devices such as hot-plugged interfaces or an

external module. The exact description and implementation of such mechanisms is out of the scope of the standard. The MISF subscribes individually with each of these link layers based on user preferences.

6.3.2.3 MIS events subscription

MIS users specify a list of events for which they wish to receive notifications from the MISF. For an MIS event that is originated either locally or remotely, an MIS user specifies whether it is subscribing for the local event only, remote event only, or both (which would require two separate subscriptions). If the MIS event that an MIS user wants to subscribe to is not supported or is not available, then the MISF rejects the subscription request and notifies the MIS user accordingly.

6.3.3 Event service flow model

Figure 14 shows the event flow model for Link events and MIS events.

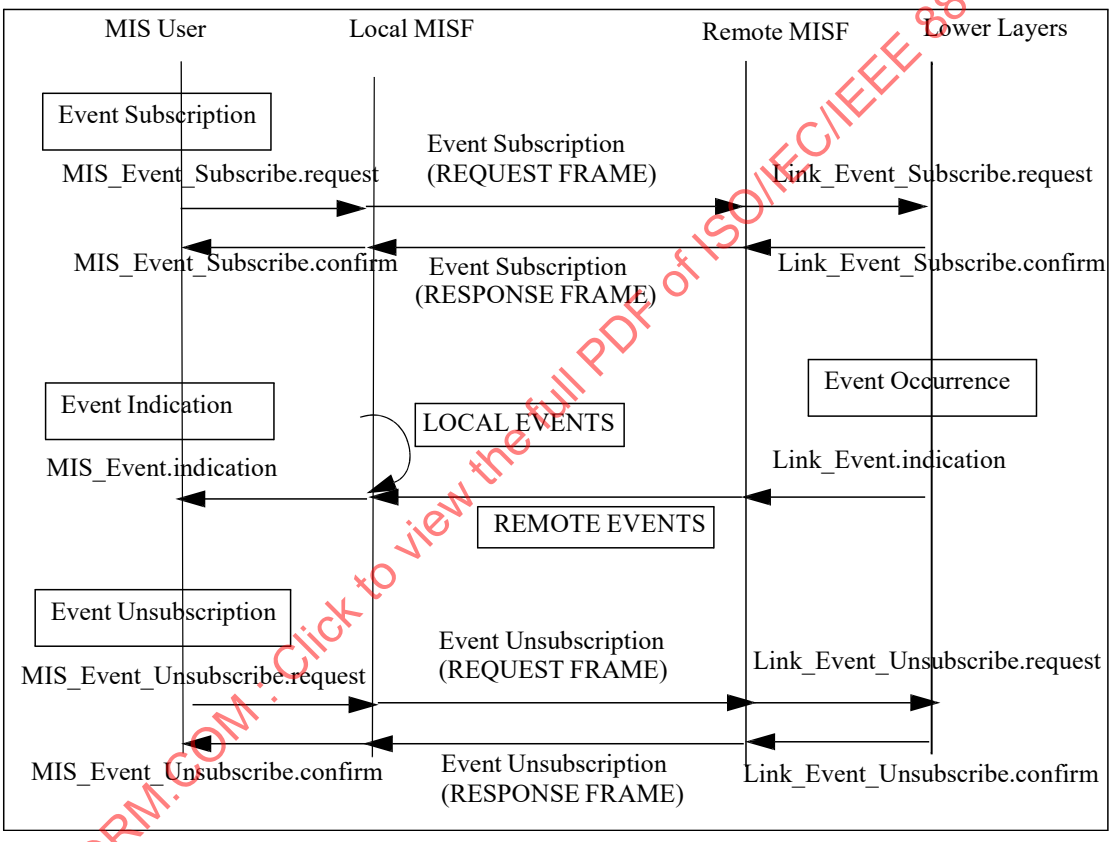


Figure 14—MIS events subscription and flow

6.3.4 Link events

The media independent event service supports the following several categories of Link events:

a) **MAC and PHY State Change events:** These events correspond to changes in MAC and PHY state. For example, Link_Up event is a state change event.

b) **Link Parameter events:** These events are due to changes in link-layer parameters. For example, the primitive Link_Parameters_Report is a Link Parameter event.

c) **Predictive events:** Predictive events convey the likelihood of a change in the link conditions in the near future based on past and present conditions. For example, decay in signal strength of a wireless local area network (WLAN) indicates a loss of link connectivity in the near future.

d) **Link Transmission events:** These events indicate the link-layer transmission status (e.g., success or failure) of upper layer PDUs. This information is used by upper layers to improve buffer management for minimizing the upper layer data loss due to a handover.

For example, the occurrence of a handover of an MN from one access network to another results in the tear-down of the old link-layer connection between the MN and the source access network and the establishment of a new link-layer connection between the MN and the target access network. When this occurs, some upper layer PDUs still remain buffered at the old link—including PDUs that had been queued at the old link but never been transmitted before the link was torn down (i.e., unsent PDUs), and PDUs that have been transmitted over the old link but never been fully acknowledged by the upper layer receiver before the link was torn down (i.e., unacknowledged PDUs). These buffered PDUs are discarded when the old link is torn down. As a result, unless the upper layer sender attempts to retransmit them over the new link connection, these upper layer PDUs never reach the receiver.

Table 4 defines Link events.

Table 4—Link events

Link event name	Link event type	Description	Defined in
Link_Detected	State change	Link of a new access network has been detected. This event is typically generated on the MN when the first PoA of an access network is detected. This event is not generated when subsequent PoAs of the same access network are discovered.	7.3.1
Link_Up	State change	L2 connection is established and link is available for use. This event is a discrete event.	7.3.2
Link_Down	State change	L2 connection is broken and link is not available for use. This event is a discrete event.	7.3.3
Link_Parameters_Report	Link parameters	Link parameters have crossed pre-specified thresholds.	7.3.4
Link_Going_Down	Predictive	Link conditions are degrading and connection loss is imminent.	7.3.5
Link_PDU_Transmit_Status	Link transmission	Indicate transmission status of a PDU.	7.3.6

In general, when a Link event occurs due to a change in link condition, it is not known at that instant if this would lead to intra-technology handover or inter-technology handover. That determination is done higher up in the protocol stack by the network selection entity based on variety of other factors. As such certain link-layer events such as Link_Going_Down leads to either intra-technology or inter-technology handovers. The network selection entity tries to maintain the current connection, by first trying intra-technology handovers and only later resort to inter-technology handovers.

6.3.5 MIS events

Table 5 defines MIS events. An MIS event is marked as local only (L), remote only (R), or local and remote (L, R), indicating whether it is subscribed by a local MIS user, a remote MIS user, or both, respectively.

Table 5—MIS events

MIS event name	(L)ocal (R)emote	Description	Defined in
MIS_Link_Detected	L, R	Link of a new access network has been detected. This event is typically generated on the MN when the first PoA of an access network is detected. This event is not generated when subsequent PoAs of the same access network are discovered.	7.4.6
MIS_Link_Up	L, R	L2 connection is established and link is available for use.	7.4.7
MIS_Link_Down	L, R	L2 connection is broken and link is not available for use.	7.4.8
MIS_Link_Parameters_Report	L, R	Link parameters have crossed a specified threshold and need to be reported.	7.4.9
MIS_Link_Going_Down	L, R	Link conditions are degrading and connection loss is imminent.	7.4.10
MIS_Link_PDU_Transmit_Status	L	Indicate transmission status of a PDU.	7.4.11

6.3.6 Interaction between MIS events and access routers

Access router (AR) is a layer 3 (L3) IP router residing in an access network and is connected to one or more PoAs. An AR is the first hop router for an MN.

During heterogeneous handovers an MN switches from one link technology to another. This results in a change in the PoA that the MN is connected to. The target PoA and the source PoA are not necessarily required to be on the same subnet. In cases where there is a change in subnet, IP packet delivery can be optimized if context (e.g., change in routing information) from the old AR to the target AR is transferred. In such cases, the target router updates its L2 address to IP address mapping.

Link-layer triggers such as Link Going Down and Link Up are used to indicate departure and arrival of MNs at AR(s) and such indications can replace L3 protocol signaling for the same and thus expedite the handover process. Layer 3 Mobility-management protocols, such as MIP, also benefit from triggers such as Link Going Down. Timely receipt of such triggers by the AR in case of network-controlled handovers enables MIP signaling to establish the new route to take place in parallel with other handover message exchange, and thus reduces the disruption time in IP packet delivery.

6.4 Media independent command service

6.4.1 Introduction

Media independent command service (MICS) refers to the commands sent from MIS users to the lower layers in the reference model. MIS users utilize command services to determine the status of links and/or control the multi-mode device for optimal performance. Command services also enable MIS users to facilitate optimal handover policies. For example, the network initiates and controls handovers to balance the load of two different access networks.

When a command request or indication frame is sent to a group of MISF peers, it is transmitted using multicast transport and one or more remote MISF(s) may receive the frame. When the frame is a command request, each recipient shall answer with a command response frame. When the frame is a command indication, no command response frame shall be returned by any recipient. An exception is that when unicast transport is used for a two-member group where one member of the group is the sender of the message and the other member is the recipient.

The link status varies with time and MN mobility. Information provided by MICS is dynamic information composed of link parameters such as signal strength and link speed; whereas, information provided by MIIS is less dynamic, or static in nature, and is composed of parameters such as network operators and

higher layer service information. MICS and MIIS information could be used in combination by the MN/network to facilitate the handover.

A number of commands are defined in this standard to allow the MIS users to configure, control, and retrieve information from the lower layers including MAC, radio-resource management, and PHY. The commands are classified into two categories: MIS commands and link commands. Figure 15 shows link commands and MIS commands.

The receipt of certain MIS command requests can cause event indications to be generated. The receipt of MIS command requests indicates a future state change in one of the link layers in the local node. These indications notify subscribed MIS users of impending link state changes. This allows MIS users to be better prepared to take appropriate action.

Link commands originate from the MISF and are directed to the lower layers. These commands mainly control the behavior of the lower layer entities. Link commands are local only. Whenever applicable, this standard encourages use of existing media-specific link commands for interaction with specific access networks. New link commands, if required, are defined as recommendations to different link-layer technology standards. It is to be noted that although link commands originate from the MISF, these commands are executed on behalf of the MIS users.

The MIS commands are generated by the MIS users and sent to the MISF. MIS commands are either local or remote. Local MIS commands are sent by MIS users to the MISF in the local protocol stack. Generally, remote commands generate an appropriate response frame from a remote MIS user, however, there are certain remote commands that do not (cf. downlink only [DO] technology related MIS commands).

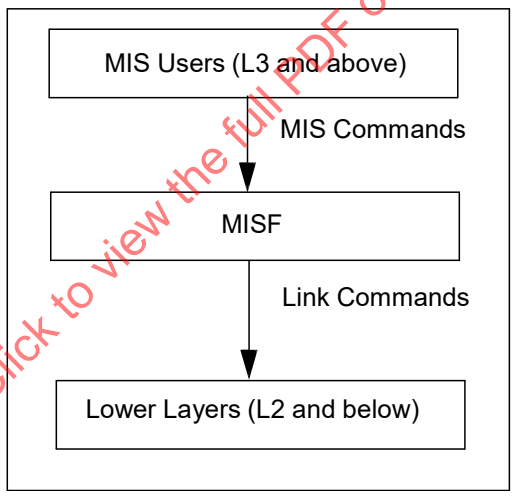


Figure 15—Link commands and MIS commands

Remote MIS commands are sent by MIS users to the MISF in a peer protocol stack. A remote MIS command delivered to a peer MISF is executed by the lower layers under the peer MISF as a link command; or is executed by the peer MISF itself as an MIS command (as if the MIS command came from an MIS user of the peer MISF); or is executed by an MIS user of the peer MISF in response to the corresponding indication. Often, an MIS indication to a remote MIS user results from the execution of the MIS command by the peer MISF. Figure 16 shows remote MIS commands.

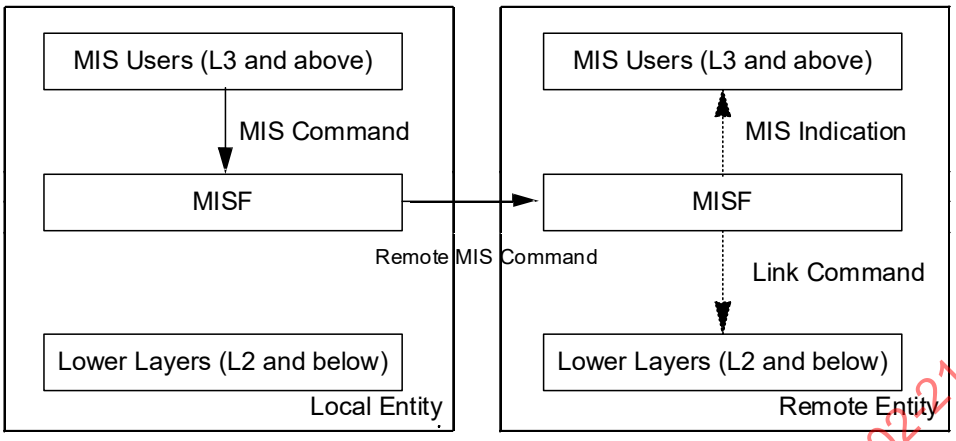
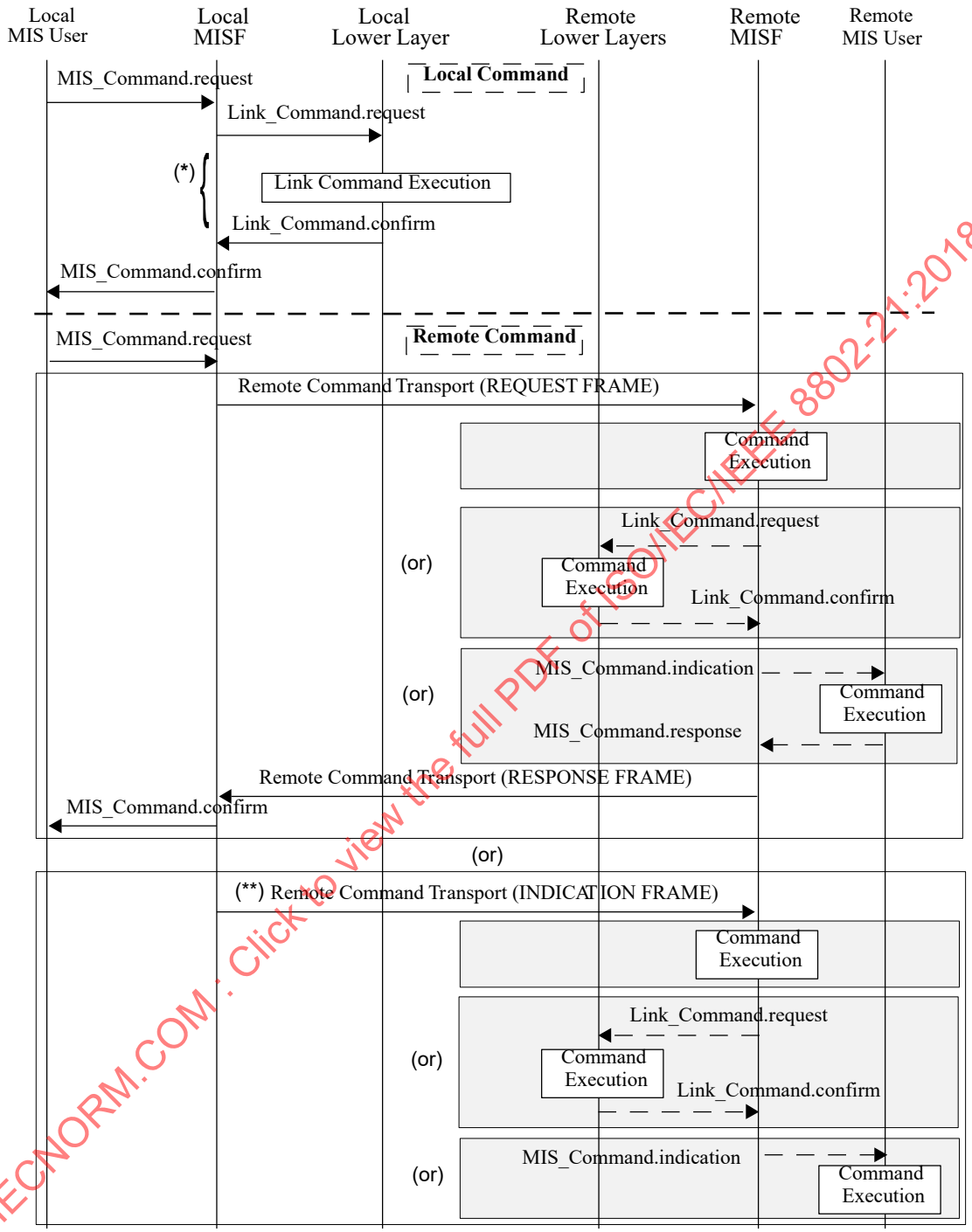


Figure 16—Remote MIS command

6.4.2 Command service flow model

Figure 17 shows the flow for a local command and an example of a remote command, respectively. Example handover procedures using the commands defined in 6.4.3 can be found in Annex B in IEEE Std 802.21.1-2017. Remote commands are transported over network layer protocols or link-layer protocols.

When a command request or indication frame is sent to a group of MISF peers, it is transmitted using multicast transport and one or more remote MISF(s) may receive the frame. When the frame is a command request, each recipient shall answer with a command response frame. When the frame is a command indication, no command response frame shall be returned by any recipient. An exception is that when unicast transport is used for a two-member group where one member of the group is the sender of the message and the other member is the recipient.



(*) There might be no corresponding Link_Command primitives, and one or more media-specific link primitives can be used here.
(**) This message applies for downlink-only technologies.

Figure 17—Command service flow

6.4.3 Command list

6.4.3.1 Link commands

Table 6 defines link commands.

Table 6—Link commands

Link command	Comments	Defined in
Link_Capability_Discover	Query and discover the list of supported link-layer events and link-layer commands.	7.3.7
Link_Event_Subscribe	Subscribe to one or more events from a link.	7.3.8
Link_Event_Unsubscribe	Unsubscribe from a set of link-layer events.	7.3.9
Link_Get_Parameters	Get parameters measured by the active link, such as signal-to-noise ratio (SNR), Bit Error Rate (BER), or received signal strength indication (RSSI).	7.3.10
Link_Configure_Thresholds	Configure thresholds for Link Parameters Report event.	7.3.11
Link_Action	Request an action on a link-layer connection.	7.3.12

6.4.3.2 MIS commands

6.4.3.2.1 General

Table 7 defines MIS Commands. An MIS command is marked as local only (L), remote only (R), or local and remote (L, R), indicating whether it is issued by a local MIS user, a remote MIS user, or both, respectively.

Table 7—MIS commands

MIS command	(L)ocal, (R)emote	Comments	Defined in
MIS_Link_Get_Parameters	L, R	Get the status of a link.	7.4.12
MIS_Link_Configure_Thresholds	L, R	Configure link parameter thresholds.	7.4.13
MIS_Link_Actions	L, R	Control the behavior of a set of links.	7.4.14

6.5 Media independent information service

6.5.1 Introduction

Media independent information service (MIIS) provides a framework by which an MISF, residing in the MN or in the network, discovers and obtains network information within a geographical area to facilitate network selection and handovers. The objective is to acquire a global view of all the heterogeneous networks relevant to the MN in the area to facilitate seamless handover across these networks.

MIIS includes support for various information elements (IEs). IEs provide information that is essential for a network selector to make intelligent handover decisions.

Depending on the type of mobility, support for different types of information elements is required for performing handovers. MIIS provides the capability for obtaining information about lower layers such as neighbor maps and other link-layer parameters, as well as information about available higher layer services such as Internet connectivity.

MIIS provides a generic mechanism to allow a service provider and a mobile user to exchange information on different handover candidate access networks. The handover candidate information includes different access technologies such as IEEE 802 networks, 3GPP networks, and 3GPP2 networks. The MIIS also allows this collective information to be accessed from any single network. For example, by using an IEEE 802.11 access network, the MN gets information not only about all other IEEE 802-based networks in a particular region, but also about 3GPP and 3GPP2 networks. Similarly, by using a 3GPP2 interface, the MN gets access to information about all IEEE 802 and 3GPP networks in a given region. This capability allows the MN to use its currently active access network and inquire about other available access networks in a geographical region. Thus, an MN is freed from the burden of powering up each of its individual radios and establishing network connectivity for the purpose of retrieving heterogeneous network information. MIIS enables this functionality across all available access networks by providing a uniform way to retrieve heterogeneous network information in any geographical area.

The main goal behind the Information Service is to allow MN and network entities to discover information that influences the selection of appropriate networks during handovers. This information is intended to be primarily used by a policy engine entity that makes effective handover decisions based on this information. This Information Service provides mostly static information, although network configuration changes are also accounted for. Other dynamic information about different access networks, such as current available resource levels, state parameters, and dynamic statistics should be obtained directly from the respective access networks. Some of the key motivations behind the Information Service are as follows:

a) Provide information about the availability of access networks in a geographical area. Further, this information could be retrieved using any wireless network; for example, information about a nearby Wi-Fi hotspot could be obtained using a global system for mobile communication (GSM), code division multiple access (CDMA), or any other cellular network, whether by means of request/response signaling, or by means of information that is specifically or implicitly broadcast over those cellular networks. Alternatively, this information could be maintained in an internal database on the MN.

b) Provide static link-layer information parameters that help the mobile nodes in selecting the appropriate access network. For example, knowledge of whether security and QoS are supported on a particular access network influences the decision to select such an access network during handovers.

c) Provide information about capabilities of different PoAs in neighbor reports to aid in configuring the radios optimally (to the extent possible) for connecting to available or selected access networks. For example, knowing about supported channels by different PoAs helps in configuring the channels optimally as opposed to scanning or beaconing and then finding out this information. Dynamic link-layer parameters have to be obtained or selected based on direct interaction with the access networks.

d) Provide an indication of higher layer services supported by different access networks and core networks that can aid in making handover decisions. Such information is not available directly from the MAC sublayer or PHY of specific access networks, but possible to be provided as part of the Information Service. For example, classification of different networks into categories, such as public, enterprise, home, and others, influences a handover decision. These higher layer services information is more vendor specific in nature.

6.5.2 Access information service before authentication

It is important to note that, with certain access networks, an MN should be able to obtain IEEE 802.21-related information elements before the MN is authenticated with the PoA. These information elements are used by the handover policy function to determine if the PoA can be selected. In order to enable the information query before authentication, individual link technologies provide an L2 or media-specific transport or a protocol message exchange that makes this MIIS query exchange possible between the user equipment (UE) and a certain MISF in the network. It should be noted that the pre-authentication query facility is provided only for MIS information query and should not be used for carrying other MIS protocol

services except MISF capability discovery query using MIS_Capability_Discover embedded into media-specific management frames. Additionally, any MISF within the network is able to request for the set of information elements from a peer MISF located in the same or a different network using the MIS protocol.

Allowing access of information service before authentication carries certain security risks, such as denial-of-service attacks and exposure of information to unauthorized MNs. In such scenarios the information service provider limits the scope of information accessible to an unauthenticated MN.

After authentication and attachment to a certain PoA, the MIS protocol is used for information retrieval by use of data frames specific to that media technology.

6.5.3 Restricting query response size

When sending an information query request, the MIIS client provides a maximum response size to limit the query response message size. A request can contain multiple queries. If the request contains multiple queries, they are in the order of significance to the client. In case the query results exceed the maximum response size, the least significant query results should be removed from the response. The MIIS server has its own maximum response size limit configured that is smaller than the one specified by the MIIS client request. In this case, the response message returns results in the order of significance to the client up to that limit.

6.5.4 Information elements

The Information Service elements are classified into the following three groups:

a) General Information and Access Network Specific Information: These information elements give a general overview of the different networks providing coverage within an area. For example, a list of available networks and their associated operators, roaming agreements between different operators, cost of connecting to the network and network security, and quality of service capabilities.

b) PoA Specific Information: These information elements (IEs) provide information about different PoAs for each of the available access networks. These IEs include PoA addressing information, PoA location, data rates supported, the type of PHY and MAC layers, and any channel parameters to optimize link-layer connectivity. This also includes higher layer services and individual capabilities of different PoAs.

c) PoS Specific Information: The information server provides access to the Point of Service information, the mobile node information, and the capability for supporting single radio handover (SRHO) for each of the available access networks. The PoS sends information elements that include PoS addressing information and tunnel management protocol information. Other information that is access network specific, service specific, or vendor/network specific.

Table 8 lists information element containers (see 6.5.6.2.1 for detailed definitions). The containers are only used in the type-length-value (TLV) based query method.

Table 8—Information element containers

Name of container	Description
IE_CONTAINER_LIST_OF_NETWORKS	List of neighboring Access Network Containers, containing information that depicts a list of heterogeneous neighboring access networks for a given geographical location.
IE_CONTAINER_NETWORK	Access Network Container, containing information that depicts an access network.
IE_CONTAINER_POA	PoA Container, containing information that depicts a PoA.

Table 9 represents the list of Information Elements and their semantics. Each Information Element has an abstract data type (see Annex E for detailed definitions). The binary and resource description framework (RDF) representation of these Information Elements are described in 6.5.6.2 and 6.5.6.3, respectively. The IEs may be retrieved using TLV- or SPARQL-based query methods. The standard does not recommend or mandate the choice of either method. An IEEE 802.21 implementation that implements the MIIS shall implement at least one method. Vendors or network operators define additional IEs beyond the IEs specified in Table 9. Vendors and network operators may implement new IEs using the vendor-specific IEs. These IEs shall then be available only in vendor- or operator-specific deployments.

Table 9—Information elements

Name of information element	Description	Data type
General information elements		
IE_NETWORK_TYPE	Link types of the access networks that are available in a given geographical area.	NETWORK_TYPE
IE_OPERATOR_ID	The operator identifier for the access network/core network.	OPERATOR_ID
IE_SERVICE_PROVIDER_ID	Identifier for the service provider.	SP_ID
IE_COUNTRY_CODE	Indicate the country.	CNTRY_CODE
Access network specific information elements		
IE_NETWORK_ID	Identifier for the access network.	NETWORK_ID
IE_NETWORK_AUX_ID	An auxiliary access network identifier. As an example, for IEEE 802.11 this refers to the homogenous extended service set ID (HESSID).	NET_AUX_ID
IE_ROAMING_PARTNERS	Roaming Partners. Network Operators with which the current network operator has direct roaming agreements.	ROAMING_PTNS
IE_COST	Cost. Indication of cost for service or network usage.	COST
IE_NETWORK_QOS	QoS characteristics of the link layer.	QOS_LIST
IE_NETWORK_DATA_RATE	Data Rate. The maximum value of the data rate supported by the link layer of the access network.	DATA_RATE
IE_NET_REGULAT_DOMAIN	Regulatory classes supported by the access network.	REGU_DOMAIN
IE_NET_FREQUENCY_BANDS	Frequency bands supported by the network.	FREQ_BANDS
IE_NET_IP_CFG_METHODS	IP Configuration Methods supported by the access network.	IP_CONFIG
IE_NET_CAPABILITIES	Bitmap of access network capabilities.	NET_CAPS
IE_NET_SUPPORTED_LCP	List of location configuration protocols supported by the access network.	SUPPORTED_LCP
IE_NET_EMSERV_PROXY	Address of the proxy providing access to public safety answering point (PSAP).	PROXY_ADDR
IE_NET_IMS_PROXY_CSCF	Address of the proxy providing access to IMS P-CSCF.	PROXY_ADDR
PoA-specific information elements		
IE_AUTHENTICATOR_LINK_ADDR	An L2 address of the authenticator, which serves the PoA.	LINK_ADDR
IE_POA_LINK_ADDR	Link-layer address of PoA.	LINK_ADDR
IE_POA_LOCATION	Geographical location of PoA. Multiple location types are supported including coordinate-based location information, civic address, and cell ID.	LOCATION
IE_POA_CHANNEL_RANGE	Channel Range/Parameters. Spectrum range supported by the channel for that PoA.	CH_RANGE
IE_POA_SYSTEM_INFO	System information supported by the link layer of a given PoA.	SYSTEM_INFO

Table 9—Information elements *(continued)*

Name of information element	Description	Data type
PoA-specific higher layer service information elements		
IE_AUTHENTICATOR_IP_ADDR	The IP address of the authenticator, which serves the PoA.	IP_ADDR
IE_POA_SUBNET_INFO	Information about subnets supported by a typical PoA.	IP_SUBNET_INFO
IE_POA_IP_ADDR	IP Address of PoA.	IP_ADDR
IE_PoS_IP_ADDR	PoS/s IP Address.	IP_ADDR
Other information elements		
Vendor-specific IEs	Vendor-specific services.	not applicable (N/A)

In certain access network deployments, some PoA properties (e.g., data rate, IP configuration methods, and capabilities) are common for all PoAs within that access network. In such a case, the common PoA properties are represented as IEs as part of the access network property information.

As an example, Figure 18 shows the layout of different Information Elements and the neighbor map of different networks in a geographical area. Multiple operators for example, Operator_1 and Operator_2, can have the same access network support, such as IEEE 802.11 network, in a geographical area. Similarly, a single operator can also provide support for multiple access networks. For example, Operator_1 provides support for IEEE 802.11 and universal mobile telecommunications system (UMTS) networks while Operator_3 provides support for IEEE 802.16 and UMTS networks. The General Network Information Elements are specified for each network supported by an operator. Thus in the case of Operator_1, General Network Information is specified for both IEEE 802.11 and UMTS networks, while in the case of Operator_2 it is specified only for an IEEE 802.11 network.

For each network supported by an operator there is a list of supported PoAs. For each PoA, the PoA Information Elements are specified. Figure 18 shows this information representation and tree hierarchy for different networks.

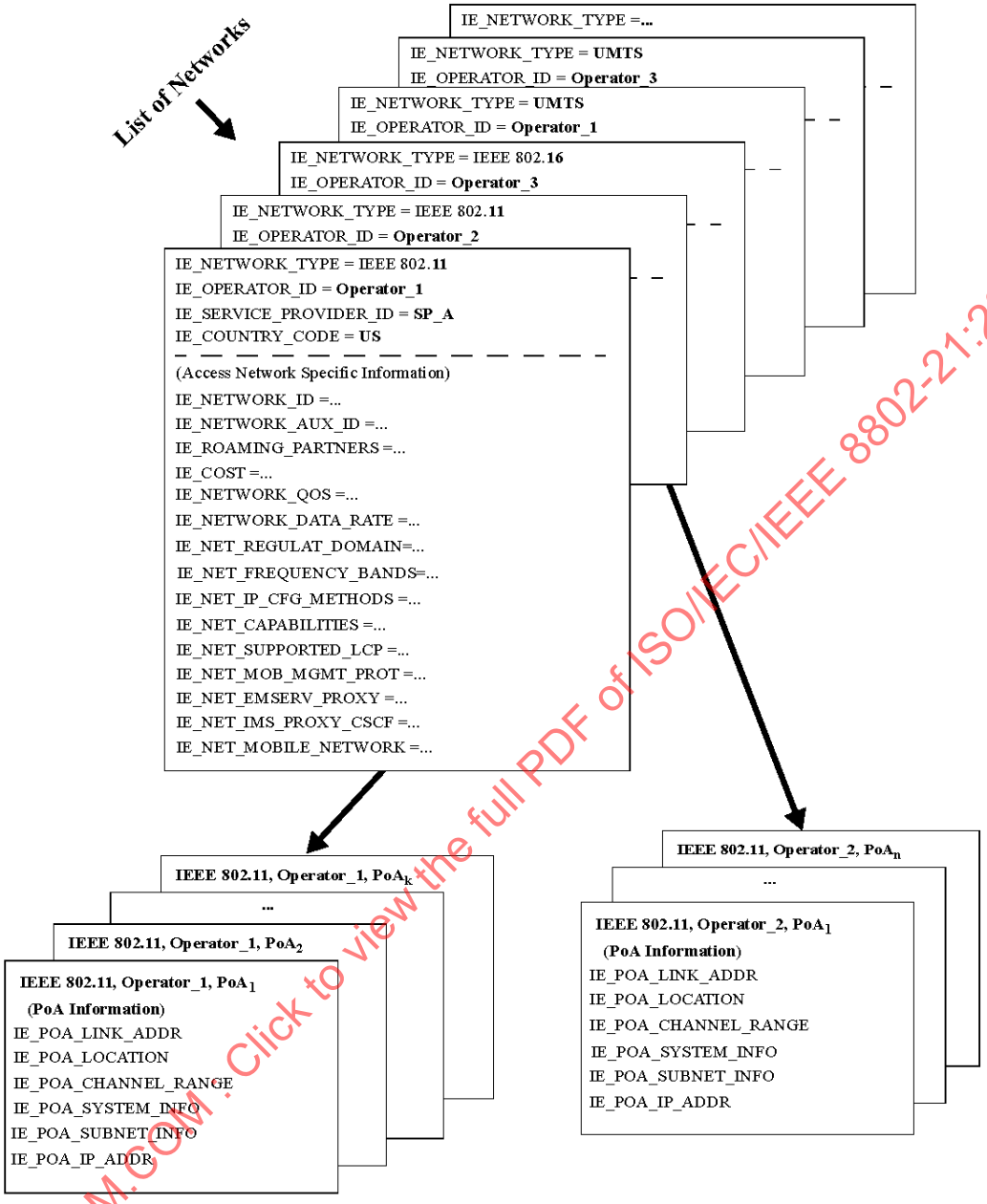


Figure 18—Depicting a list of neighboring networks with information elements

6.5.5 Definition of information element namespace

Each Information Element ID is a 32-bit value. Table 10 defines the Information Element namespace. The IEEE 802.21 specific Information Elements are assigned identifiers as per this standard. Please refer to Table F.1 for more details. Vendors specify their own IEs using the namespace allocated to them. A set of IE namespace ranges is also reserved for development and testing. These should not be used in released products. Allocation of additional IE namespace and any revisions to this assignment will be handled by future revisions of this standard.

Table 10—Information element namespace

Range	Description	Comments
0x00000000	Reserved	
0x00000001–0x1FFFFFFF	IEEE 802.21 IEs	Used for IEEE 802.21 defined IEs. The currently defined IEEE 802.21 IEs are listed in Table F.1.
0x20000000–0x7FFFFFFF	Vendor-specific IEs	Used for IEs defined by vendors. To prevent vendor-specific IE collisions, the 2nd, 3rd, and 4th octet are filled with the value of the vendor's IEEE organizationally unique identifier (OUI[a]) or Company ID (CID[a]). For example, if a vendor's IEEE OUI is 00-03-3F, then its corresponding vendor-specific IE range would be 0x2000033F–0x7F00033F.
0x80000000–0x82FFFFFF	Reserved for playpen area	Used in development and testing. Should not be used in released products. Avoids collision during development.
0x83000000–0xFFFFFFFF	Reserved	For future use.

[a] Interested applicants should contact the IEEE Registration Authority, http://standards.ieee.org/regauth.

Functional entities should discard any received IE with an unrecognizable identifier.

6.5.6 Information element representation and query methods

6.5.6.1 Introduction

MIIS defines two methods for representing Information Elements: binary representation and RDF representation (see W3C Recommendation, Resource Description Framework [RDF]–Concepts and Abstract Syntax and W3C Recommendation, RDF/XML Syntax Specification). MIIS also defines two query methods. For requests using the binary representation, the TLV query method defined in 6.5.6.2 is used. For requests using the RDF representation, the SPARQL (see W3C Recommendation, SPARQL Query Language for RDF) query method is used.

6.5.6.2 Binary representation and TLV query

In the binary representation method, information elements are represented and encoded in type-length-value (TLV) form as shown in Figure 19.

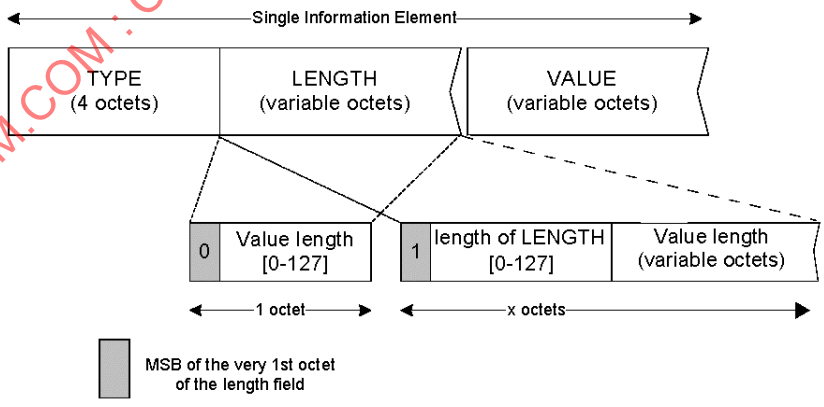


Figure 19—TLV representation of information elements

The *Length* field is interpreted as follows:

Case 1: If the number of octets occupied by the *Value* field is less than 128, the size of the *Length* field is always one octet and the MSB of the octet is set to the value '0'. The values of the other seven bits of this octet indicate the actual length of the *Value* field.

Case 2: If the number of octets occupied by the *Value* field is exactly 128, the size of the *Length* field is one octet. The MSB of the *Length* octet is set to the value '1' and the other seven bits of this octet are all set to the value '0'.

Case 3: If the number of octets occupied by the *Value* field is greater than 128, then the *Length* field is always greater than one octet. The MSB of the first octet of the *Length* field is set to the value '1' and the remaining seven bits of the first octet indicate the number of octets that are appended further. The number represented by the second and subsequent octets of the *Length* field, when added to 128, indicates the total size of the *Value* field, in octets.

6.5.6.2.1 IE containers

In the binary representation method, three Information Element Containers are defined, namely the IE_CONTAINER_LIST_OF_NETWORKS, the IE_CONTAINER_NETWORK, and the IE_CONTAINER_POA:

— **IE_CONTAINER_LIST_OF_NETWORKS**—contains a list of heterogeneous neighboring access networks for a given geographical location, as shown in Table 11.

An IE_CONTAINER_LIST_OF_NETWORKS contains at least one Access Network and optionally one or more vendor-specific IEs. When more than one Access Network Container is provided in this IE, they should be prioritized in the order of preference from the information server's perspective with first Access Network Container as the top priority and with decreasing priority going down the list. This would enable the receiving entity to utilize this information in the same way as provided in this list for network selection or handover decisions.

Table 11—IE_CONTAINER_LIST_OF_NETWORKS definition

Information element ID = (see Table F.1)	**Length** = *variable*
IE_CONTAINER_NETWORK #1	
IE_CONTAINER_NETWORK #2 (optional)	
…	
IE_CONTAINER_NETWORK #k (optional)	
Vendor-specific IE (optional)	

— **IE_CONTAINER_ NETWORKS**—contains all the information depicting an access network, as shown in Table 12.

When more than one PoA Container is provided in this IE, they should be prioritized in the order of preference from the information server's perspective with first PoA Container as the top priority and with decreasing priority going down the list. This would enable the receiving entity to utilize this information in the same way as provided in this list for network selection or handover decision.

Table 12—IE_CONTAINER_NETWORKS definition

Information element ID = (see Table F.1)	**Length** = *variable*
IE_NETWORK_TYPE	
IE_OPERATOR_ID	
IE_SERVICE_PROVIDER_ID (optional)	
IE_COUNTRY_CODE (optional)	
IE_NETWORK_ID (optional)	
IE_NETWORK_AUX_ID (optional)	
IE_ROAMING_PARTNERS (optional)	
IE_COST (optional)	
IE_NETWORK_QOS (optional)	
IE_NETWORK_DATA_RATE (optional)	
IE_NET_REGULAT_DOMAIN (optional)	
IE_NET_FREQUENCY_BANDS (optional)	
IE_NET_IP_CFG_METHODS (optional)	
IE_NET_CAPABILITIES (optional)	
IE_NET_SUPPORTED_LCP (optional)	
IE_NET_EMSERV_PROXY (optional)	
IE_NET_IMS_PROXY_CSCF (optional)	
IE_CONTAINER_POA #1 (optional)	
IE_CONTAINER_POA #2 (optional)	
…	
IE_CONTAINER_POA #k (optional)	
Vendor-specific network IE (optional)	

— **IE_CONTAINER_POA**—contains all the information depicting a PoA and optionally one or more vendor-specific PoA IEs, as shown in Table 13.

Table 13—IE_CONTAINER_POA definition

Information element ID = (see Table F.1)	**Length** = *variable*
IE_POA_LINK_ADDR	
IE_POA_LOCATION	
IE_POA_CHANNEL_RANGE	
IE_POA_SYSTEM_INFO	
IE_POA_SUBNET_INFO #1	
IE_POA_SUBNET_INFO #2 (optional)	
...	
IE_POA_SUBNET_INFO #k (optional)	
IE_POA_IP_ADDR #1 (optional)	
...	
IE_POA_IP_ADDR #k (optional)	
Vendor-specific PoA IE (optional)	

TLVs for the component IEs contained in the Access Network Container and PoA Container are defined in Annex E.

6.5.6.2.2 TLV queries

A TLV query includes the following optional parameters to refine the query.

QUERIER_LOC parameter (defined in Table E.15) is used for the information server to refine its response. The value field contains either the querier's current location measurement or, when the querier does not have its current location information, an observed link-layer address (e.g., from an IEEE 802.11 Beacon frame or some broadcast mechanism for other technologies) that the information server should be able to use as a hint to establish an estimate of the client's current location. Within the QUERIER_LOC parameter, the querier should not use both the LINK_ADDR value (defined in) and LOCATION value (defined in Table E.10) in the same query. Moreover, the NGHB_RADIUS value (defined in Table E.15), if provided, indicates the radius of the neighborhood, centered at the indicated location, within which all available access networks are included in the list of neighboring networks. If NGHB_RADIUS value is not present, the information server should decide the radius for the search.

If QUERIER_LOC parameter is not included in the query, the information server either gets the querier location information through other means or uses the best estimate of the querier's location to generate the neighboring network information.

NET_TYPE_INC parameter (see Table E.15 for definition) is used to indicate the neighboring network types the querier wants to include in the response. The querier indicates the network types it wants to include in the query response by setting the corresponding bits to '1.' If not provided, the information server includes information about all available network types in the query response.

NETWK_INC parameter (see Table E.15 for definition) is used to indicate the specific access networks the querier wants to include in the query response. If not provided, the information server includes information about all available access networks in the query response.

RPT_TEMPL parameter (see Table E.15 for definition) is used to give the information server a template of the list of IEs that is included in the information response.

The following rules shall be followed for using RPT_TEMPL parameter:

— If the RPT_TEMPL parameter is absent, the entire list of neighboring networks container is returned in the response (subject to constraints on message length, as defined in 6.5.3).

— If a container is listed *without* any of its component IEs, the entire container is returned in the response (subject to constraints on message length, as defined in 6.5.3). For example, inclusion of *IE_CONTAINER_POA* solely returns a list of PoA Containers with all their component IEs.

— If a container is listed *with* one or more of its component IEs, the container *with only* the listed component IEs is returned. For example, inclusion of *IE_CONTAINER_NETWORK, IE_NETWORK_TYPE,* and *IE_OPERATOR_ID* solely returns a list of Network Containers with each containing only Network Type and Operator ID.

— If a component IE is listed *without* its parent container, the listed component IE is returned as an individual IE. For example, inclusion of *IE_NETWORK_TYPE* and *IE_COST* solely returns a list of Network Types and a list of Costs.

NOTE—A list of individual IEs out of their context has very limited usefulness. This is only an example to show the flexible use of RPT_TEMPL parameter.[13]

The following rules are followed for generating returned IEs:

Upon receipt of a binary query, the information server shall:

a) Create the list of neighboring access network information for the given location.

   1) If a NET_TYPE_INC parameter is provided in the query, include only the information of the neighboring access networks of the network type(s) indicated in the NET_TYPE_INC parameter. Otherwise, include information of all available neighboring access networks for the given location.

   2) If a NETWK_INC parameter is provided in the query, include only the information of the neighboring access network(s) indicated in the NETWK_INC parameter. Otherwise, include information of all available neighboring access networks for the given location.

b) If no RPT_TEMPL parameter is given in the query, send the list of neighboring access network information in an IE_CONTAINER_LIST_OF_NETWORKS in an MIS_Get_Information response message.

[13] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

c) If an RPT_TEMPL parameter is given in the query, extract the requested IE(s)/Containers from the list of neighboring access network information using the rules described for RPT_TEMPL parameter and send them in an MIS_Get_Information response message.

6.5.6.3 RDF representation and SPARQL query

The RDF representation of Information Elements is represented in XML format. SPARQL is used as the query method. The RDF representation and SPARQL query method implement the RDF schema as described in 6.5.7.2.

6.5.7 Information service schema

6.5.7.1 General

A schema is used in the IEEE 802.21 Information Service to define the structure of each information element, as well as the relationship among the information elements. The IEEE 802.21 Information Service schema is supported by every MISF that implements the MIIS to support flexible and efficient information queries.

6.5.7.2 MIIS RDF schema

The RDF schema definition for MIIS consists of two parts; the basic and the extended schema. An MIIS client or server should be pre-provisioned with the basic schema for ease of implementation of schema-based query. In scenarios where the basic schema is not pre-provisioned, methods such as dynamic host configuration protocol (DHCP) are used to obtain the basic schema.

The MIIS RDF representation method is extensible using the extended schema. The extended schema can be pre-provisioned. The extended schema can also be updated dynamically, e.g., when a new information element about the network is introduced. When the extended schema is not pre-provisioned, it is retrieved from the specified uniform resource locator (URL) via the IEEE 802.21 Information Service using the schema query capability. Alternatively, methods such as DHCP provide the URL of the extended schema as well. The implementation should always use the updated version of extended schema as opposed to using the pre-provisioned version.

The basic schema is defined in Annex G. The basic schema contains the schema for information elements defined in Table 9. The extended schema is defined by individual vendors or by network operators and contain the schema for vendor-specific information elements or network operator–specific information. (See Annex H for an example of a vendor-specific extension.)

6.5.8 Information service flow

Figure 20 describes an Information Service flow. The MIIS within an MISF communicates with the remote MISF that resides within the access network. MIS_Get_Information from the MN is carried over the appropriate transport (L2 or L3) and is delivered to the remote MISF. The remote MISF returns the necessary information to the MN via the appropriate response frame.

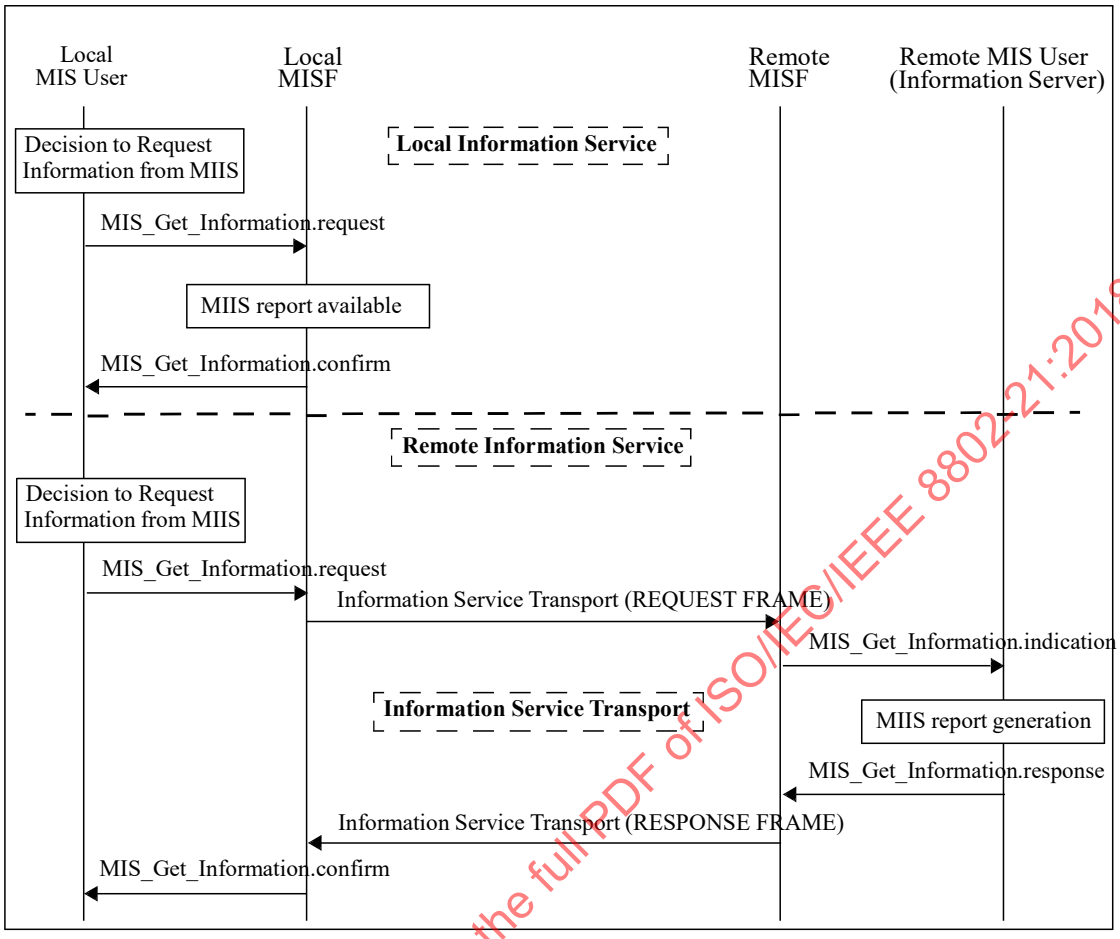


Figure 20—MIIS information flow

7. Service access point (SAPs) and primitives

7.1 Introduction

The MIS function uses the following SAPs for interfacing with other entities.

Media dependent SAPs:

a) MIS_LINK_SAP: Abstract media dependent interface of MISF with the lower layers of the media-specific protocol stacks. The mappings between MIS_LINK_SAP and various media-specific SAPs are described in D.2.

b) MIS_NET_SAP: Abstract media dependent interface of MISF that provides transport services over the data plane on the local node, supporting the exchange of MIS information and messages with the remote MISF.

Media independent SAP:

— MIS_SAP: This SAP defines the media independent interface between the MISF and MIS users.

7.2 SAPs

7.2.1 General

The SAPs are defined as a set of primitives. Taken together, the primitives define the services. Within the definition of each primitive there is a table of allowable parameters. Each parameter is defined using abstract data types. These types indicate the semantic value of that parameter. The parameters defined within the subclause for a particular primitive are produced or consumed by that primitive. Several of the abstract data types are used in multiple primitive definitions. In each abstract data type definition, the various names applied to this type are listed in Annex E.

7.2.2 Media dependent SAPs

7.2.2.1 MIS_LINK_SAP

The primitives defined as part of the MIS_LINK_SAP are described in Table 14. Annex D contains their mapping to several specific link technologies. IETF RFC 5184 specifies many of these primitives as L2 abstractions.

Table 14—MIS_LINK_SAP primitives

Primitives	Service category	Description	Defined in
Link_Detected	Event	A new link is detected.	7.3.1
Link_Up	Event	L2 connectivity is established.	7.3.2
Link_Down	Event	L2 connectivity is lost.	7.3.3
Link_Parameters_Report	Event	Link parameters have crossed specified thresholds.	7.3.4
Link_Going_Down	Event	L2 connectivity loss is imminent.	7.3.5
Link_PDU_Transmit_Status	Event	Indicate transmission status of a PDU.	7.3.6
Link_Capability_Discover	Command	Query and discover the list of supported link-layer events and link-layer commands.	7.3.7
Link_Event_Subscribe	Command	Subscribe for event notifications.	7.3.8
Link_Event_Unsubscribe	Command	Unsubscribe from event notifications.	7.3.9
Link_Get_Parameters	Command	Request parameters of medium.	7.3.10
Link_Configure_Thresholds	Command	Configure link thresholds for Link events.	7.3.11
Link_Action	Command	Request an action on a link-layer connection.	7.3.12

7.2.2.2 MIS_NET_SAP

The primitive defined for MIS_NET_SAP is described in Table 15.

Table 15—MIS_NET_SAP primitive

Primitive	Service category	Description	Defined in
MIS_TP_Data	Network communication	This primitive is used for transfer of data.	7.5.1

7.2.3 Media independent SAP: MIS_SAP

The primitives defined as part of MIS_SAP are described in Table 16.

Table 16—MIS_SAP primitives

Primitives	Service category	Description	Defined in
MIS_Capability_Discover	Service management	Discover list of Events and Commands supported by MISF.	7.4.1
MIS_Register	Service management	Register with a remote MISF.	7.4.2
MIS_DeRegister	Service management	Deregister with a remote MISF.	7.4.3
MIS_Event_Subscribe	Service management	Subscribe for MIS event notifications.	7.4.4
MIS_Event_Unsubscribe	Service management	Unsubscribe from MIS event notifications.	7.4.5
MIS_Link_Detected	Event	A new link is detected.	7.4.6
MIS_Link_Up	Event	L2 connection has been established.	7.4.7
MIS_Link_Down	Event	L2 connectivity is lost.	7.4.8
MIS_Link_Parameters_Report	Event	Link parameters have crossed specified threshold.	7.4.9
MIS_Link_Going_Down	Event	L2 connectivity is predicted to go down.	7.4.10
MIS_Link_PDU_Transmit_Status	Event	Indicate transmission status of a PDU.	7.4.11
MIS_Link_Get_Parameters	Command	Get the status of link.	7.4.12
MIS_Link_Configure_Thresholds	Command	Configure link parameter thresholds.	7.4.13
MIS_Link_Actions	Command	Control the behavior of a set of links.	7.4.14
MIS_Get_Information	Information	Request to get information from repository.	7.4.15
MIS_Push_Information	Information	Notify the mobile node of operator policies or other information.	7.4.16
MIS_Push_Key	Service management	Install a key in a remote PoA.	7.4.17
MIS_LL_Auth	Service management	Carry out a proactive authentication over MIS messages between the MN and the PoS using link-layer frames.	7.4.18
MIS_Configuration_Update	Service management	Updates the configuration of one or more MN(s) or PoS(s).	7.4.19
MIS_Pull_Group_Manipulate	Service management	Used by an MN or a PoS to manipulate its group membership.	7.4.20
MIS_Push_Group_Manipulate	Service management	Used by a group manager to manipulate the group membership of a node.	7.4.21
MIS_Pull_Certificate	Service management	Request the sending of a certificate from the destination PoS to the requestor.	7.4.22
MIS_Push_Certificate	Service management	Used to send a certificate to a destination PoS or MN.	7.4.23
MIS_Revoke_Certificate	Service management	Used to revoke a certificate.	7.4.24

MIS command primitives defined in MIS_SAP indicate their destination as either the local MISF or a remote MISF. For the remote case, the local MISF first processes the primitive to create an MIS message and then forwards the message to the destination peer MISF for execution. In those messages, there are

TLV-encoded parameters that implement the primitive parameter abstract data types within the protocol. The definition of the full binary encoding for each of these instantiations is in Annex E.

7.3 MIS_LINK_SAP primitives

7.3.1 Link_Detected.indication

7.3.1.1 Function

Link_Detected indicates the presence of a new PoA. This implies that the MN is in the coverage area. Link_Detected does not guarantee that the MN is able to establish connectivity with the detected link, but just that the MN identified the link to gain connectivity. MIS users and the MISF evaluate additional properties of the link before attempting to establish an L2 connection with the link. Moreover, Link_Detected is not generated when additional PoAs of the same link are discovered. In case of IEEE 802.11, Link_Detected is generated by MAC state generic convergence function (MSGCF).

7.3.1.2 Semantics of service primitive

Link_Detected.indication (
LinkDetectedInfo
)

Parameters:

Name	Data type	Description
LinkDetectedInfo	LINK_DET_INFO	Information of a detected link.

7.3.1.3 When generated

The Link Detected event is generated on the MN when the first PoA of an access network is detected. This event is not generated when subsequent PoAs of the same access network are discovered during the active connection on that link.

7.3.1.4 Effect on receipt

The MISF receives this event from the link layer. The MISF shall pass this notification to the MIS user(s) that has subscribed for this notification. The MIS user(s), including the MISF itself, discovers additional properties of the link before selecting it for establishing connectivity.

7.3.2 Link_Up.indication

7.3.2.1 Function

This notification is delivered when a layer 2 connection is established on the specified link interface. All layer 2 activities in establishing the link connectivity are expected to be completed at this point in time.

7.3.2.2 Semantics of service primitive

Link_Up.indication (
LinkIdentifier,
OldAccessRouter,
NewAccessRouter,
IPRenewalFlag
)

Parameters:

Name	Data type	Description
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
OldAccessRouter	LINK_ADDR	(Optional) Old Access Router link address.
NewAccessRouter	LINK_ADDR	(Optional) New Access Router link address.
IPRenewalFlag	IP_RENEWAL_FLAG	(Optional) Indicates whether the MN needs to change IP Address in the new PoA.

7.3.2.3 When generated

This notification is generated when a layer 2 connection is established for the specific link interface.

7.3.2.4 Effect on receipt

The MISF shall pass this link notification to the MIS user(s) that has subscribed for this notification in an MIS_Link_Up event. The MIS user(s) takes different actions on this notification.

7.3.3 Link_Down.indication

7.3.3.1 Function

This notification is delivered when a layer 2 connection is no longer available for sending frames, that is, when the L2 connection with network is terminated and not during PoA to PoA transitions for the same network.

7.3.3.2 Semantics of service primitive

Link_Down.indication (
LinkIdentifier,
OldAccessRouter,
ReasonCode
)

Parameters:

Name	Data type	Description
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
OldAccessRouter	LINK_ADDR	(Optional) Old Access Router link address.
ReasonCode	LINK_DN_REASON	Reason why the link went down.

7.3.3.3 When generated

This notification generated when layer 2 connectivity is lost. Layer connectivity is lost explicitly in cases where the MN initiates disassociate type procedures. In other cases, the MN can infer loss of link connectivity due to successive time-outs for acknowledgements of transmitted packets along with loss of reception of broadcast frames.

7.3.3.4 Effect on receipt

The MISF passes this link notification to the MIS user(s) that has subscribed for this notification in an MIS_Link_Down event. The MIS user(s) takes different actions on this notification. The handover policy

function can eliminate this link from list of active links for routing connections and can consider handing over any potential active connections to other more suitable links.

7.3.4 Link_Parameters_Report.inidication

7.3.4.1 Function

Link_Parameters_Report indicates changes in link conditions that have crossed specified threshold levels. Link_Parameters_Report is also generated at specified intervals for various parameters.

In the case of IEEE 802.11 network, this event is generated when higher protocol layers wish to monitor the performance parameters for a network. In some cases, these higher layers are on the network side (for network initiated handovers) and MISF on the local MN transfers these parameters. For local MN initiated handovers, the local station management entity (SME) and MSGCF would monitor link-layer properties and the MISF would normally be interested only in the Link_Going_Down.indication.

NOTE—The primitive to set parameter thresholds that could trigger this event is specified in 7.3.11.

7.3.4.2 Semantics of service primitive

Link_Parameters_Report.indication (
LinkIdentifier,
LinkParametersReportList
)

Parameters:

Name	Data type	Description
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
LinkParametersReportList	LIST(LINK_PARAM_RPT)	A list of Link Parameter Report.

7.3.4.3 When generated

For each specified parameter, this notification is generated either at a predefined regular interval determined by a user configurable timer or when it crosses a configured threshold.

7.3.4.4 Effect on receipt

The MISF receives this event from the link layer. The MISF then passes this notification to the MIS user(s) that has subscribed for this notification. The MIS user(s) takes different actions on this notification. If parameters related to link quality cross a certain threshold, then that link needs to be evaluated for handing over current connections. The MISF collectively evaluates different parameters and gives appropriate indications to higher layers regarding suitability of different links.

7.3.5 Link_Going_Down.indication

7.3.5.1 Function

This notification is delivered when a Layer 2 connection is expected (predicted) to go down (Link_Down) within a certain time interval. Link_Going_Down event is an indication to initiate handover procedures.

7.3.5.2 Semantics of service primitive

Link_Going_Down.indication (
LinkIdentifier,
TimeInterval,
LinkGoingDownReason
)

Parameters:

Name	Data type	Description
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
TimeInterval	UNSIGNED_INT(2)	Time Interval (in milliseconds) specifies the time interval at which the link is expected to go down. A value of '0' is specified if the time interval is unknown.
LinkGoingDownReason	LINK_GD_REASON	The reason why the link is going to be down.

7.3.5.3 When generated

A Link_Going_Down event implies that a Link_Down is imminent within a certain time interval. If Link_Down is NOT received within specified time interval, then actions due to previous Link_Going_Down are ignored.

In the case of IEEE 802.11 networks, this notification is generated when the established IEEE 802.11 network connection is expected to go down within the specified time interval by the IEEE 802.11 MSGCF. The network is expected to go down because of an event whose timing is well understood, such as an explicit disconnection event observed on the MLME SAP. This is expected as the result of a predictive algorithm that monitors the link quality. The details of such a predictive algorithm used are beyond the scope of this standard. This event is not generated when the IEEE 802.11 station (STA) transitions from one AP to another in the same network.

7.3.5.4 Effect on receipt

The MISF receives this event from the link layer. The MISF then passes this notification to the MIS user(s) that has subscribed for this notification. MIS user(s) takes different actions on this notification. MIS users, then, prepare to initiate handovers.

7.3.6 Link_PDU_Transmit_Status.indication

7.3.6.1 Function

Link_PDU_Transmit_Status indicates the transmission status of a higher layer PDU by the link layer. A success status indicates that the higher layer PDU has been successfully delivered from the link layer in the local node to the link layer in the peer node. A higher layer intermediate buffer management entity could use this indication to flush the delivered PDU from its buffer. A failure status indicates that the higher layer PDU identified in the indication was not delivered successfully from the link layer in the local node to the link layer in the peer node. During a handover, if such a failure indication is received from the link connection with the source network, the higher layer intermediate buffer management entity could attempt to retransmit the failed PDU once a connection to the target network is established.

A Packet Identifier is expected to be passed alongside when each higher layer PDU is sent from the higher layer to the link for transmission. The Packet Identifier is defined in this standard as a container structure whose syntax and semantics should be decided by the upper layer (i.e., the MIS user that subscribes to this

event). The MISF and link layer just pass and return the Packet Identifier and do not need to understand its syntax and semantics.

To avoid receiving an excessive amount of link PDU transmission status indications, an MIS user, for example, chooses to subscribe to this event only after it receives a Link_Handover_Imminent.indication or when it is about to invoke an MIS_Link_Actions.request to perform a handover, and to unsubscribe from the event once it receives indication that the handover is completed.

7.3.6.2 Semantics of service primitive

Link_PDU_Transmit_Status.indication (
LinkIdentifier,
PacketIdentifier,
TransmissionStatus
)

Parameters:

Name	Data type	Description
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
PacketIdentifier	UNSIGNED_INT(2)	Identifier for higher layer PDU on which this notification is generated.
TransmissionStatus	BOOLEAN	Status of the transmitted packet. TRUE: Success FALSE: Failure

7.3.6.3 When generated

A success notification is generated when a higher layer PDU is successfully transmitted over the link. A failure notification is generated when a higher layer PDU was not transmitted successfully.

7.3.6.4 Effect on receipt

The MISF receives this event from the link layer. The MISF then passes this notification to the MIS user(s) that has subscribed for this notification. The MIS user(s) takes different actions on this notification. A higher layer intermediate buffer management entity in MIS could use the success indication to flush higher layer packets stored in any intermediate buffers and a failure indication to retransmit higher layer packets stored in any intermediate buffers, especially if there are changes in the access network during handovers.

7.3.7 Link_Capability_Discover

7.3.7.1 Link_Capability_Discover.request

7.3.7.1.1 Function

This primitive is used by the MISF to query and discover the list of supported link-layer events and link-layer commands.

7.3.7.1.2 Semantics of service primitive

No primitive parameters exist for this primitive.

Link_Capability_Discover.request ()

7.3.7.1.3 When generated

This primitive is generated by the MISF when it needs to receive link-layer event notifications and learn about which link-layer commands the lower layer is able to support.

7.3.7.1.4 Effect on receipt

The recipient responds immediately with Link_Capability_Discover.confirm primitive.

7.3.7.2 Link_Capability_Discover.confirm

7.3.7.2.1 Function

This primitive returns the result of the query to discover link-layer capability.

7.3.7.2.2 Semantics of service primitive

Link_Capability_Discover.confirm (
Status,
SupportedLinkEventList,
SupportedLinkCommandlist,
SupportedLinkActionsList
)

Parameters:

Name	Data type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
SupportedLinkEventList[a]	LINK_EVENT_LIST	List of link-layer events supported by the link layer.
SupportedLinkCommandList[a]	LINK_CMD_LIST	List of link-layer commands supported by the link layer.
SupportedLinkActionsList[a]	SUPPORTED_LINK_ACTIONS_LIST	(Optional) This optional parameter is present if bit 5 of SupportedLinkCommandList is set to 1, SupportedLinkActionsList indicates which link actions are supported by the link.

[a]This parameter is not included if Status does not indicate "Success."

7.3.7.2.3 When generated

This primitive is generated in response to a Link_Capability_Discover.request primitive.

7.3.7.2.4 Effect on receipt

The recipient examines the returned event and command list and learns about link-layer capability. However, if Status does not indicate "Success," the recipient performs appropriate error handling.

7.3.8 Link_Event_Subscribe

7.3.8.1 Link_Event_Subscribe.request

7.3.8.1.1 Function

This primitive is used by MISF (the subscriber) to subscribe an interest in one or more events from a specific link-layer technology. The response indicates which of the requested events were successfully subscribed to. Events that were not successfully subscribed to shall not be delivered to the subscriber.

7.3.8.1.2 Semantics of service primitive

Link_Event_Subscribe.request (
RequestedLinkEventList
)

Parameter:

Name	Data type	Description
RequestedLinkEventList	LINK_EVENT_LIST	List of link-layer events that for which the subscriber would like to receive indications.

7.3.8.1.3 When generated

This primitive is generated by a subscriber such as the MISF that is seeking to receive event indications from different link-layer technologies.

7.3.8.1.4 Effect on receipt

The recipient responds immediately with Link_Event_Subscribe.confirm primitive.

7.3.8.2 Link_Event_Subscribe.confirm

7.3.8.2.1 Function

This primitive returns the result of the subscription request.

7.3.8.2.2 Semantics of service primitive

Link_Event_Subscribe.confirm (
Status,
ResponseLinkEventList
)

Parameters:

Name	Data type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
ResponseLinkEventList[a]	LINK_EVENT_LIST	List of successfully subscribed Link events.

[a]This parameter is not included if Status does not indicate “Success.”

7.3.8.2.3 When generated

This primitive is generated in response to a Link_Event_Subscribe.request primitive.

7.3.8.2.4 Effect on receipt

The recipient examines the ResponseLinkEventList and learns about the subscription status of different events. If Status does not indicate "Success," the recipient performs appropriate error handling.

7.3.9 Link_Event_Unsubscribe

7.3.9.1 Link_Event_Unsubscribe.request

7.3.9.1.1 Function

This primitive is used by the MIHF to query and discover the list of supported link-layer events and linklayer commands.

7.3.9.1.2 Semantics of service primitive

Link_Event_Unsubscribe.request (
RequestedLinkEventList
)

Parameter:

Name	Data type	Description
RequestedLinkEventList	LINK_EVENT_LIST	List of link-layer events for which indications need to be unsubscribed from the Event Source.

7.3.9.1.3 When generated

This primitive is generated by a subscriber such as the MISF that is seeking to unsubscribe from an already subscribed set of events.

7.3.9.1.4 Effect on receipt

The recipient responds immediately with Link_Event_Unsubscribe.confirm primitive.

7.3.9.2 Link_Event_Unsubscribe.confirm

7.3.9.2.1 Function

This primitive returns the result of the request to unsubscribe from receiving link-layer event notifications.

7.3.9.2.2 Semantics of service primitive

Link_Event_Unsubscribe.confirm (
Status,
ResponseLinkEventList
)

Parameters:

Name	Data type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
ResponseLinkEventList[a]	LINK_EVENT_LIST	List of successfully unsubscribed Link events.

[a]This parameter is not included if Status does not indicate "Success."

7.3.9.2.3 When generated

The primitive is generated in response to a Link_Event_Unsubscribe.request primitive.

7.3.9.2.4 Effect on receipt

The recipient should examine the ResponseLinkEventList and learn about the unsubscription status of different events. If Status does not indicate "Success." The recipient performs appropriate error handling.

7.3.10 Link_Get_Parameters

7.3.10.1 Link_Get_Parameters.request

7.3.10.1.1 Function

The primitive is used by the MISF to obtain the current value of a set of link parameters of a specific link.

7.3.10.1.2 Semantics of service primitive

Link_Get_Parameters.request (
LinkParametersRequest,
LinkStatesRequest,
LinkDescriptorsRequest
)

Parameters:

Name	Data type	Description
LinkParametersRequest	LIST(LINK_PARAM_TYPE)	A list of link parameters for which status is requested.
LinkStatesRequest	LINK_STATES_REQ	The link states to be requested.
LinkDescriptorsRequest	LINK_DESC_REQ	The link descriptors to be requested.

7.3.10.1.3 When generated

The primitive is generated by the MISF to obtain the current value of a set of link parameters from a link.

7.3.10.1.4 Effect on receipt

The recipient link responds with Link_Get_Parameters.confirm primitive.

7.3.10.2 Link_Get_Parameters.confirm

7.3.10.2.1 Function

This primitive is sent in response to the Link_Get_Parameters.request primitive. This primitive provides current value of the requested link parameters.

NOTE—How the value is measured or calculated by the link is not specified by this standard.

7.3.10.2.2 Semantics of service primitive

Link_Get_Parameters.confirm (
Status,
LinkParametersStatusList,
LinkStatesResponse,
LinkDescriptorsResponse
)

Parameters:

Name	Data type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
LinkParametersStatusList[a]	LIST(LINK_PARAM)	A list of measurable link parameters and their current values.
LinkStatesResponse[a]	LIST(LINK_STATES_RSP)	The current link state information.
LinkDescriptorsResponse[a]	LIST(LINK_DESC_RSP)	The descriptors of a link.

[a]This parameter is not included if Status does not indicate "Success."

7.3.10.2.3 When generated

This primitive is generated in response to the Link_Get_Parameters.request operation.

7.3.10.2.4 Effect on receipt

The recipient passes the link parameter values received to the MIS users. However, if Status does not indicate "Success," the recipient performs appropriate error handling.

7.3.11 Link_Configure_Thresholds

7.3.11.1 Link_Configure_Thresholds.request

7.3.11.1.1 Function

This primitive is used by the MISF to configure thresholds and/or specify the time interval between periodic reports for the Link_Parameters_Report indication.

7.3.11.1.2 Semantics of service primitive

Link_Configure_Thresholds.request (
LinkConfigureParameterList
)

Parameter:

Name	Data type	Description
LinkConfigureParameterList	LIST(LINK_CFG_PARAM)	A list of link threshold parameters.

7.3.11.1.3 When generated

This primitive is generated by an MISF that needs to set threshold values for different link parameters.

7.3.11.1.4 Effect on receipt

The recipient responds immediately with Link_Configure_Thresholds.confirm primitive.

7.3.11.2 Link_Configure_Thresholds.confirm

7.3.11.2.1 Function

This primitive is sent in response to the Link_Configure_Thresholds.request primitive. This primitive specifies the status of threshold configuration operation.

7.3.11.2.2 Semantics of service primitive

Link_Configure_Thresholds.confirm (
Status,
LinkConfigureStatsList
)

Parameters:

Name	Data type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
LinkConfigureStatusList[a]	LIST(LINK_CFG_STATUS)	A list of Link Configure Status.

[a]This parameter is not included if Status does not indicate “Success.”

7.3.11.2.3 When generated

This primitive is generated in response to the Link_Configure_Thresholds.request operation.

7.3.11.2.4 Effect on receipt

The recipient prepares to receive Link_Parameters_Report indications on successful execution of this primitive. However, if Status does not indicate “Success,” the recipient performs appropriate error handling.

7.3.12 Link_Action

7.3.12.1 Link_Action.request

7.3.12.1.1 Function

The primitive is used by the MISF to request an action on a link-layer connection to enable optimal handling of link-layer resources for the purpose of handovers.

The link-layer connection should be ordered (e.g., to shut down, to remain active, to perform a scan, or to come up active and remain in stand-by mode). The command execution delay time should also be specified for cases where the link-layer technology under consideration supports the action.

7.3.12.1.2 Semantics of service primitive

Link_Action.request (
LinkAction,
ExecutionDelay,
PoAlinkAddress
)

Parameters:

Name	Data type	Description
LinkAction	LINK_ACTION	Specifies the action to perform.
ExecutionDelay	UNSIGNED_INT(2)	Time (in ms) to elapse before the action needs to be taken. A value of 0 indicates that the action is taken immediately. Time elapsed is calculated from the instant the request arrives until the time when the execution of the action is carried out.
PoALinkAddress	LINK_ADDR	(Optional) The PoA link address to forward data to. This parameter is used when DATA_FWD_REQ action is requested.

7.3.12.1.3 When generated

The MISF generates this primitive upon request from the MIS user to perform an action on a pre-defined link-layer connection.

7.3.12.1.4 Effect on receipt

Upon receipt of this primitive, the link-layer technology supporting the current link-layer connections performs the action specified by the LinkAction parameter in accordance with the procedures specified by the relevant standards organization and at the time specified by the Execution Delay parameter.

7.3.12.2 Link_Action.confirm

7.3.12.2.1 Function

This primitive is used by link-layer technologies to provide an indication of the result of the action executed on the current link-layer connection.

7.3.12.2.2 Semantics of service primitive

Link_Action.confirm (
Status,
ScanResponseSet,
LinkActionResult
)

Parameters:

Name	Data type	Description
Status	STATUS	Status of the operation. Code 3 (Authorization Failure) is not applicable.

ScanResponseSet[a]	LIST(LINK_SCAN_RSP)	(Optional) A list of discovered links and related information.
LinkActionResult[a]	LINK_AC_RESULT	Specifies whether the link action was successful.

[a]This parameter is not included if Status does not indicate "Success."

7.3.12.2.3 When generated

The link-layer technology generates this primitive to communicate the result of the action executed on the link-layer connection.

7.3.12.2.4 Effect on receipt

Upon receipt of this primitive, the MISF determines the relevant MIS command that needs to be used to provide an indication or confirmation to the MIS user of the actions performed on the current link-layer connection. If a Scan action was issued by the associated Link_Action.request, the optional ScanResponseSet field is included in the Link_Action.confirm response.

7.4 MIS_SAP primitive

7.4.1 MIS_Capability_Discover

7.4.1.1 MIS_Capability_Discover.request

7.4.1.1.1 Function

This primitive is used by an MIS user to discover the capabilities of the local MISF or a remote MISF. When invoking this primitive to discover the capabilities of a remote MISF, the MIS user optionally piggybacks the capability information of its local MISF so that the two MISFs mutually discover each other's capabilities with a single invocation of this primitive.

7.4.1.1.2 Semantics of service primitive

MIS_Capability_Discover.request (
DestinationIdentifier,
LinkAddressList,
SupportedMISEventList,
SupportedMISCommandList,
SupportedISQueryTypeList,
SupportedTransportList,
SupportedSecurityCapList,
SupportedLinkActionsList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies the local MISF or a remote MISF that shall be the destination of this request.
LinkAddressList	LIST(NET_TYPE_ADDR)	(Optional) A list of network type and link address pair on the local MISF.
SupportedMISEventList	MIS_EVT_LIST	(Optional) List of supported events on the local MISF.
SupportedMISCommandList	MIS_CMD_LIST	(Optional) List of supported commands on the local MISF.

SupportedISQueryTypeList	MIS_IQ_TYPE_LST	(Optional) List of supported MIIS query types on the local MISF.
SupportedTransportList	MIS_TRANS_LST	(Optional) List of supported transport types on the local MISF.
SupportedSecurityCapList	MIS_SEC_CAP	(Optional) List of supported MIS security capabilities on the local MISF.
SupportedLinkActionsList	SUPPORTED_LINK_ACTIONS_LIST	(Optional) This optional parameter is present if bit 2 of SupportedMISCommandList is set to 1. SupportedLinkActionsList indicates the list of supported link actions on the local MISF.

7.4.1.1.3 When generated

This primitive is generated by an MIS user to discover the capabilities of the local MISF or a remote MISF. In the case of remote discovery, this primitive contains the SupportedMISEventList, SupportedMISCommandList, SupportedISQueryTypeList, and SupportedTransportList parameters of the local MISF to enable mutual discovery of each other's capabilities.

7.4.1.1.4 Effect on receipt

If the destination of the request is the local MISF itself, the local MISF responds with MIS_Capability_Discover.confirm. If the destination of the request is a remote MISF, the local MISF shall generate a corresponding MIS_Capability_Discover request message to the remote MISF if it does not have the capability information of the remote MISF.

7.4.1.2 MIS_Capability_Discover.indication

7.4.1.2.1 Function

This primitive is used by an MISF to notify an MIS user on the receipt of an MIS_Capability_Discover request message from a peer MISF.

7.4.1.2.2 Semantics of service primitive

MIS_Capability_Discover.indication (
SourceIdentifier,
LinkAddressList,
SupportedMISEventList,
SupportedMISCommandList,
SupportedISQueryTypeList,
SupportedTransportList,
SupportedSecurityCapList,
SupportedLinkActionsList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is a remote MISF.
LinkAddressList	LIST(NET_TYPE_ADDR)	(Optional) A list of network type and link address pair on the remote MISF.
SupportedMISEventList	MIS_EVT_LIST	(Optional) List of supported events on the remote MISF.

SupportedMISCommandList	MIS_CMD_LIST	(Optional) List of supported commands on the remote MISF.
SupportedISQueryTypeList	MIS_IQ_TYPE_LST	(Optional) List of supported MIIS query types on the remote MISF.
SupportedTransportList	MIS_TRANS_LST	(Optional) List of supported transport types on the remote MISF.
SupportedSecurityCapList	MIS_SEC_CAP	(Optional) List of supported MIS security capabilities on the remote MISF.
SupportedLinkActionsList	SUPPORTED_LINK_ACTIONS_LIST	(Optional) This optional parameter is present if bit 2 of SupportedMISCommandList is set to 1, SupportedLinkActionsList indicates the list of supported link actions on the remote MISF.

7.4.1.2.3 When generated

This primitive is used by an MISF to notify an MIS user when an MIS_Capability_Discover request message is received. This primitive is optional since the MISF is capable of returning and an immediate MIS_Capability_Discover response message without generating this primitive to the MIS user.

7.4.1.2.4 Effect on receipt

The MIS user responds with an MIS_Capability_Discover.response primitive when an indication is received.

7.4.1.3 MIS_Capability_Discover.response

7.4.1.3.1 Function

This primitive is used by an MIS user to convey the locally supported MIS capabilities to the MIS user that invoked the MIS_Capability_Discover request.

7.4.1.3.2 Semantics of Service primitive

MIS_Capability_Discover.response (
DestinationIdentifier,
Status,
LinkAddressList,
SupportedMISEventList,
SupportedMISCommandList,
SupportedISQueryTypeList,
SupportedTransportList,
SupportedSecurityCapList,
SupportedLinkActionsList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies the remote MISF that shall be the destination of this response.
Status	STATUS	Status of operation.
LinkAddressList	LIST(NET_TYPE_ADDR)	(Optional) A list of network type and link address pair on local MISF.

SupportedMISEventList	MIS_EVT_LIST	(Optional) List of supported events on local MISF.
SupportedMISCommandList	MIS_CMD_LIST	(Optional) List of supported commands on local MISF.
SupportedISQueryTypeList	MIS_IQ_TYPE_LST	(Optional) List of supported MIIS query types on local MISF.
SupportedTransportList	MIS_TRANS_LST	(Optional) List of supported transport types on local MISF.
SupportedSecurityCapList	MIS_SEC_CAP	(Optional) List of supported MIS security capabilities on the local MISF.
SupportedLinkActionsList	SUPPORTED_LINK_ACTIONS_LIST	(Optional) This optional parameter is present if bit 2 of SupportedMISCommandList is set to 1, SupportedLinkActionsList indicates the list of supported link actions on the local MISF.

7.4.1.3.3 When generated

This primitive is generated by an MIS user as a response to a received MIS_Capability_Discover.indication primitive.

7.4.1.3.4 Effect on receipt

Upon receiving this primitive, the MISF shall generate and send the corresponding MIS_Capability_Discover response message to the destination MISF.

7.4.1.4 MIS_Capability_Discover.confirm

7.4.1.4.1 Function

This primitive is used by the MISF to convey the supported MIS capabilities about Event Service, Command Service, and Information Service to the MIS user that invoked the MIS_Capability_Discover.request.

7.4.1.4.2 Semantics of service primitive

MIS_Capability_Discover.confirm (
SourceIdentifier,
Status,
LinkAddressList,
SupportedMISEventList,
SupportedMISCommandList,
SupportedISQueryTypeList,
SupportedTransportList,
SupportedSecurityCapList,
SupportedLinkActionsList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
Status	STATUS	Status of operation.
LinkAddressList	LIST(NET_TYPE_ADDR)	(Optional) A list of network type and link address pair on the MISF identified by Source Identifier.

SupportedMISEventList	MIS_EVT_LIST	(Optional) List of supported events on the MISF identified by Source Identifier.
SupportedMISCommandList	MIS_CMD_LIST	(Optional) List of supported commands on the MISF identified by Source Identifier.
SupportedISQueryTypeList	MIS_IQ_TYPE_LST	(Optional) List of supported MIIS query types on the MISF identified by Source Identifier.
SupportedTransportList	MIS_TRANS_LST	(Optional) List of supported transport types on the MISF identified by Source Identifier.
SupportedSecurityCapList	MIS_SEC_CAP	(Optional) List of supported MIS security capabilities on the remote MISF.
SupportedLinkActionsList	SUPPORTED_LINK_ ACTIONS_LIST	(Optional) This optional parameter is present if bit 2 of SupportedMISCommandList is set to 1, SupportedLinkActionsList indicates the list of supported link actions on the MISF identified by SourceIdentifier.

7.4.1.4.3 When generated

This primitive is invoked by a local MISF to convey the results of a previous MIS_Capability_Discover.request primitive from an MIS user.

7.4.1.4.4 Effect on receipt

Upon reception of this primitive the receiving entity becomes aware of the supported MIS capabilities. However, if Status does not indicate "Success," the recipient ignores any other returned values and, instead, performs appropriate error handling.

7.4.2 MIS_Register

7.4.2.1 MIS_Register.request

7.4.2.1.1 Function

This primitive is used by an MIS user to register the local MISF with remote MISF.

7.4.2.1.2 Semantics of service primitive

MIS_Register.request (
DestinationIdentifier,
LinkIdentifierList,
GroupLinkIdentifier,
RequestCode
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF that shall be the destination of this request.
LinkIdentifierList	LIST(LINK_ID)	(Optional) List of link identifiers of the local MISF. This parameter shall be used if and only if DestinationIdentifier is an MISF ID.
GroupLinkIdentifier	NET_TYPE_INC	(Optional) Identifier of a group of links. This parameter shall be used if and only if DestinationIdentifier is an MISF Group ID.

RequestCode	REG_REQUEST_CODE	Registration request code. Depending on the request code, the MIS user chooses to either register or re-register with the remote MISF.

7.4.2.1.3 When generated

This primitive is invoked by the MIS user when it needs to register the local MISF with a remote MISF.

7.4.2.1.4 Effect on receipt

On receipt, the local MISF sends an MIS_Register request message to the destination MISF.

7.4.2.2 MIS_Register.indication

7.4.2.2.1 Function

This primitive is used by an MISF to notify an MIS user that an MIS_Register request message has been received.

7.4.2.2.2 Semantics of service primitive

MIS_Register.indication (
SourceIdentifier,
LinkIdentifierList,
GroupLinkIdentifier,
RequestCode
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is a remote MISF.
LinkIdentifierList	LIST(LINK_ID)	(Optional) List of local link identifiers of the remote MISF. This parameter shall be used if and only if DestinationIdentifier is an MISF ID.
GroupLinkIdentifier	NET_TYPE_INC	(Optional) Identifier of a group of links. This parameter shall be used if and only if DestinationIdentifier is an MISF Group ID.
RequestCode	REG_REQUEST_CODE	Registration request code. Depending on the request code, the MIS user chooses to either register or re-register with the remote MISF.

7.4.2.2.3 When generated

This primitive is generated when an MIS_Register request message is received.

7.4.2.2.4 Effect on receipt

The MIS user performs necessary actions to process the registration request and respond with an MIS_Register.response.

7.4.2.3 MIS_Register.response

7.4.2.3.1 Function

This primitive is used by an MIS user to send the processing status of a received registration request.

7.4.2.3.2 Semantics of service primitive

MIS_Register.response (
DestinationIdentifier,
Status,
ValidTimeInterval,
MulticastCipherSuite,
Certificate
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF, which shall be the destination of this response.
Status	STATUS	Status of operation.
ValidTimeInterval[a]	UNSIGNED_INT(4)	Time interval in seconds during which the registration is valid. Parameter applicable only when the status parameter indicates a successful operation. A value of 0 indicates an infinite validity period.
MulticastCipherSuite	MULTICAST_CAP	(Optional) Specifies the group ciphersuite to be used for securing multicast MIS messages.
Certificate	CERTIFICATE	(Optional) X.509 certificate.

[a]This parameter is not included if Status does not indicate "Success."

7.4.2.3.3 When generated

This primitive is invoked by the MIS user to report back the result after completing the processing of a registration request.

7.4.2.3.4 Effect on receipt

Upon receipt, the local MISF sends an MIS_Register response message to the destination MISF.

7.4.2.4 MIS_Register.confirm

7.4.2.4.1 Function

This primitive is used by the local MISF to convey the result of a registration request to an MIS user.

7.4.2.4.2 Semantics of service primitive

MIS_Register.confirm (
SourceIdentifier,
Status,
ValidTimeInterval,
MulticastCipherSuite,
Certificate
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is a remote MISF.
Status	STATUS	Status of operation.
ValidTimeInterval[a]	UNSIGNED_INT(4)	Time interval in seconds during which the registration is valid. Parameter applicable only when the status parameter indicates a successful operation. A value of 0 indicates an infinite validity period.
MulticastCipherSuite	MULTICAST_CAP	(Optional) Specifies the group ciphersuite to be used for securing multicast MIS messages.
Certificate	CERTIFICATE	(Optional) X.509 certificate.

[a]This parameter is not included if Status does not indicate "Success."

7.4.2.4.3 When generated

This primitive is used by an MISF to notify an MIS user the result of an MIS registration request.

7.4.2.4.4 Effect on receipt

Upon receipt, the MIS user is able to determine the result of the registration request.

7.4.3 MIS_DeRegiser

7.4.3.1 MIS_DeRegister.request

7.4.3.1.1 Function

This primitive is used by an MIS user to deregister the local MISF with peer MISF.

7.4.3.1.2 Semantics of service primitive

MIS_DeRegister.request (
DestinationIdentifier
)

Parameter:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF that shall be the destination of this request.

7.4.3.1.3 When generated

This primitive is invoked by the MIS user when it needs to terminate an existing MIS registration with a remote MISF.

7.4.3.1.4 Effect on receipt

Upon receipt, the local MISF generates and sends an MIS_DeRegister request message to the destination MISF.

7.4.3.2 MIS_DeRegister.indication

7.4.3.2.1 Function

This primitive is used by an MISF to notify an MIS user that an MIS_DeRegister request message has been received.

7.4.3.2.2 Semantics of service primitive

MIS_DeRegister.indication (
SourceIdentifier
)

Parameter:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is a remote MISF.

7.4.3.2.3 When generated

This primitive is generated by an MISF when an MIS_DeRegister request message is received.

7.4.3.2.4 Effect on receipt

The MIS user performs necessary actions to process the deregistration request and respond with an MIS_DeRegister.response.

7.4.3.3 MIS_DeRegister.response

7.4.3.3.1 Function

This primitive is invoked by a remote MIS user to respond with the processing status of a received deregistration request.

7.4.3.3.2 Semantics of service primitive

MIS_DeRegister.response (
DestinationIdentifier,
Status
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF, which shall be the destination of this response.
Status	STATUS	Status of operation. Code 2 (Reject) is not used.

7.4.3.3.3 When generated

This primitive is invoked by the MIS user to report back the result after completing the processing of a deregistration request from a remote MIS user.

7.4.3.3.4 Effect on receipt

Upon receipt, the local MISF sends an MIS_DeRegister response message to the destination MISF.

7.4.3.4 MIS_DeRegister.confirm

7.4.3.4.1 Function

This primitive is used by the local MISF to convey the result of a deregistration request to the local MIS user.

7.4.3.4.2 Semantics of service primitive

MIS_DeRegister.confirm (
SourceIdentifier,
Status
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is a remote MISF.
Status	STATUS	Status of operation. Code 2 (Rejected) is not used.

7.4.3.4.3 When generated

This primitive is used by an MISF to notify the local MIS user the status of MIS deregistration request.

7.4.3.4.4 Effect on receipt

Upon receipt, the MIS user is able to determine the status of the deregistration request.

7.4.4 MIS_Event_Subscribe

7.4.4.1 MIS_Event_Subscribe.request

7.4.4.1.1 Function

This primitive is used by an MIS user (the subscriber) to subscribe an interest in one or more MIS event types from the local or a remote MISF. Optionally, the subscriber indicates a list of specific configuration information applicable for various events being subscribed. If configured, the event shall be triggered only when all the criteria set in the parameters are met.

7.4.4.1.2 Semantics of service primitive

MIS_Event_Subscribe.request (
DestinationIdentifier,
LinkIdentifier,
GroupLinkIdentifier,
RequestedMISEventList,
EventConfigurationInfoList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies the local MISF or a remote MISF that shall be the destination of this request.
LinkIdentifier	LINK_TUPLE_ID	(Optional) Identifier of the link for event subscription. For local event subscription, PoA link address need not be present if the link type lacks such a value. This parameter shall be used if and only if DestinationIdentifier is an MISF ID.
GroupLinkIdentifier	NET_TYPE_INC	(Optional) Identifier of a group of links for event subscription. This parameter shall be used if and only if DestinationIdentifier is an MISF Group ID.
RequestedMISEventList	MIS_EVT_LIST	List of MIS events that the endpoint would like to receive indications for, from the Event Source.
EventConfigurationInfoList	LIST(EVT_CFG_INFO)	(Optional) List of additional configuration information for event subscription.

7.4.4.1.3 When generated

This primitive is invoked by an MIS user when it wants to receive indications on a set of specific MIS events from the local MISF or a remote MISF.

7.4.4.1.4 Effect on receipt

If the destination of the request is the local MISF itself, the local MISF responds immediately with an MIS_Event_Subscribe.confirm primitive. If the destination of the request is a remote MISF, the local MISF generates and sends an MIS_Event_Subscribe request message to the remote MISF.

7.4.4.2 MIS_Event_Subscribe.confirm

7.4.4.2.1 Function

This primitive returns the result of an MIS event subscription request.

7.4.4.2.2 Semantics of service primitive

MIS_Event_Subscribe.confirm (
SourceIdentifier,
Status,
LinkIdentifier,
ResponseMISEventList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
Status	STATUS	Status of operation.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link for event subscription.
ResponseMISEventList[a]	MIS_EVT_LIST	List of successfully subscribed MIS events.

[a]This parameter is not included if Status does not indicate “Success.”

7.4.4.2.3 When generated

This primitive is generated by the local MISF at the completion of processing an MIS_Event_Subscribe.request primitive from a local MIS user or in response to the receiving of an MIS_Event_Subscribe response message from a peer MISF.

7.4.4.2.4 Effect on receipt

The recipient MIS user examines the returned event list and learns about the subscription status of different events. However, if Status does not indicate "Success," the recipient performs appropriate error handling.

7.4.5 MIS_Event_Unsubscribe

7.4.5.1 MIS_Event_Unsubscribe.request

7.4.5.1.1 Function

This primitive is used by an MIS user (the subscriber) to unsubscribe from a set of previous subscribed MIS events.

7.4.5.1.2 Semantics of service primitive

MIS_Event_Unsubscribe.request (
DestinationIdentifier,
LinkIdentifier,
GroupLinkIdentifier,
RequestedMISEventList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies the local MISF or a remote MISF, which shall be the destination of this request.
LinkIdentifier	LINK_TUPLE_ID	(Optional) Identifier of the link for event unsubscription. For local event unsubscription, PoA address in the Link Identifier need not be present if the link type lacks such a value. This parameter shall be used if and only if DestinationIdentifier is an MISF ID.
GroupLinkIdentifier	NET_TYPE_INC	(Optional) Identifier of a group of links for event unsubscription. This parameter shall be used if and only if DestinationIdentifier is an MISF Group ID.
RequestedMISEventList	MIS_EVT_LIST	List of MIS events for which indications need to be unsubscribed from the Event Source.

7.4.5.1.3 When generated

This primitive is invoked by an MIS user (subscriber) that is seeking to unsubscribe from an already subscribed set of events from the local MISF or a remote MISF.

7.4.5.1.4 Effect on receipt

If the destination of the request is the local MISF itself, the local MISF responds immediately with MIS_Event_Unsubscribe.confirm primitive. If the destination of the request is a remote MISF, the local MISF generates and sends an MIS_Event_Unsubscribe request message to the remote MISF.

7.4.5.2 MIS_Event_Unsubscribe.confirm

7.4.5.2.1 Function

This primitive returns the result of an MIS event unsubscription request.

7.4.5.2.2 Semantics of service primitive

MIS_Event_Unsubscribe.confirm (
SourceIdentifier,
Status,
LinkIdentifier,
ResponseMISEventList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
Status	STATUS	Status of operation.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link for event unsubscription.
ResponseMISEventList[a]	MIS_EVT_LIST	List of successfully unsubscribed Link events.

[a]This parameter is not included if Status does not indicate "Success."

7.4.5.2.3 When generated

This primitive is generated by the local MISF at the completion of processing an MIS_Event_Unsubscribe.request primitive from a local MIS user or in response to the receiving of an MIS_Event_Unsubscribe response message from a peer MISF.

7.4.5.2.4 Effect on receipt

The recipient MIS user is able to examine the returned event list and learn about the unsubscription status of different events. However, if Status does not indicate "Success," the recipient performs appropriate error handling.

7.4.6 MIS_Link_Detected.indication

7.4.6.1 Function

The MIS_Link_Detected.indication is sent to local MISF users to notify them of a local event or of a receipt of MIS_Link_Detected indication message from a remote MISF.

7.4.6.2 Semantics of the service primitive

MIS_Link_Detected.indication (
SourceIdentifier,
LinkDetectedInfoList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
LinkDetectedInfoList	LIST(LINK_DET_INFO)	List of link detection information.

7.4.6.3 When generated

The MIS_Link_Detected.indication is sent to local MISF users to notify them of a local event (i.e., Link_Detected.indication), or of receipt of MIS_Link_Detected indication message from a remote MISF (i.e., a remote Link_Detected event has occurred).

7.4.6.4 Effect on receipt

MIS user dependant.

7.4.7 MIS_Link_Up.indication

7.4.7.1 Function

The MIS_Link_Up.indication is sent to local MISF users to notify them of a local event, or is the result of the receipt of an MIS_Link_Up indication message to indicate to the remote MISF users who have subscribed to this remote event.

7.4.7.2 Semantics of the service primitive

MIS_Link_Up.indication (
SourceIdentifier,
LinkIdentifier,
OldAccessRouter,
NewAccessRouter,
IPRenewalFlag
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
OldAccessRouter	LINK_ADDR	(Optional) Link address of old Access Router.
NewAccessRouter	LINK_ADDR	(Optional) Link address of new Access Router.
IPRenewalFlag	IP_RENEWAL_FLAG	(Optional) Indicates whether the MN needs to change IP Address in the new PoA.

7.4.7.3 When generated

The MIS_Link_Up.indication is sent to local MISF users to notify them of a local event (i.e., Link_Up.indication), or is the result of the receipt of an MIS_Link_Up indication message to indicate to the remote MISF users who have subscribed to this remote event that a remote link up event occurred.

7.4.7.4 Effect on receipt

MIS user dependant.

7.4.8 MIS_Link_Down.indication

7.4.8.1 Function

The MIS_Link_Down.indication is sent to local MISF users to notify them of a local event, or is the result of the receipt of an MIS_Link_Down indication message to indicate to the remote MISF users who have subscribed to this remote event.

7.4.8.2 Semantics of the service primitive

MIS_Link_Down.indication (
SourceIdentifier,
LinkIdentifier,
OldAccessRouter,
ReasonCode
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
OldAccessRouter	LINK_ADDR	(Optional) Link address of old Access Router.
ReasonCode	LINK_DN_REASON	Reason why the link went down.

7.4.8.3 When generated

The MIS_Link_Down.indication is sent to local MISF users to notify them of a local event (i.e., Link_Down.indication), or is the result of the receipt of an MIS_Link_Down indication message to indicate to the remote MISF users who have subscribed to this remote event that a remote link_down event occurred.

7.4.8.4 Effect on receipt

MIS user dependant.

7.4.9 MIS_Link_Parameters_Report.indication

7.4.9.1 Function

MIS_Link_Parameters_Report indication is sent by the local MISF to a local MIS user to report the status of a set of parameters of a local or remote link. This MIS event is either local or remote.

7.4.9.2 Semantics of service primitive

MIS_Link_Parameters_Report.indication (
SourceIdentifier,
LinkIdentifier,
LinkParameterReportList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
LinkParameterReportList	LIST(LINK_PARAM_RPT)	A list of Link Parameter Reports.

7.4.9.3 When generated

This notification is generated by the local MISF either

— At a predefined regular interval determined by a user configurable timer;

— When a specified parameter of a currently active local interface crosses a configured threshold. In such a case, the local MISF most likely first receives a Link_Parameters_Report.indication from the local link layer; or

— When an MIS_Link_Parameters_Report indication message is received from a remote MISF.

7.4.9.4 Effect on receipt

Upper layer entities take different actions upon receipt of this indication.

7.4.10 MIS_Link_Going_Down.indication

7.4.10.1 Function

The MIS_Link_Going_Down.indication is sent to local MISF users to notify them of a local event, or is the result of the receipt of an MIS_Link_Going_Down indication message to indicate to the remote MISF users who have subscribed to this remote event.

7.4.10.2 Semantics of the service primitive

MIS_Link_Going_Down.indication (
SourceIdentifier,
LinkIdentifier,
TimeInterval,
LinkGoingDownReason
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
TimeInterval	UNSIGNED_INT(2)	Time Interval (in milliseconds) specifies the time interval at which the link is expected to go down. A value of ‘0’ is specified if time interval is unknown or uncertain.
LinkGoingDownReason	LINK_GD_REASON	The reason why the link is going down.

7.4.10.3 When generated

The MIS_Link_Going_Down.indication is sent to local MISF users to notify them of a local event (i.e., Link_Going_Down.indication), or is the result of the receipt of an MIS_Link_Going_Down indication message to indicate to the remote MISF users who have subscribed to this remote event that a remote link_going_down event occurred.

7.4.10.4 Effect on receipt

MIS user dependant.

7.4.11 MIS_Link_PDU_Transmit_Status.indication

7.4.11.1 Function

The MIS_Link_PDU_Transmit_Status.indication is sent to local MISF users to notify them of a local event.

7.4.11.2 Semantics of the service primitive

MIS_Link_PDU_Transmit_Status.indication (
SourceIdentifier,
LinkIdentifier,
PacketIdentifier,
TransmissionStatus
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the local MISF where this event occurred.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event.
PacketIdentifier	UNSIGNED_INT(2)	Identifier for higher layer PDU on which this notification is generated.
TransmissionStatus	BOOLEAN	Status of the transmitted packet. TRUE: Success FALSE: Failure

7.4.11.3 When generated

The MIS_Link_PDU_Transmit_Status.indication is sent to local MISF users to notify them of a local event (i.e., Link_PDU_Transmit_Status.indication).

7.4.11.4 Effect on receipt

MIS user dependant.

7.4.12 MIS_Link_Get_Parameters

7.4.12.1 General

An MIS_Link_Get_Parameters command is issued by upper layer entities to discover and monitor the status of the currently connected and potentially available links. This command is also used to get device state information. The destination of an MIS_Link_Get_Parameters command is local or remote. For example, an MIS_Link_Get_Parameters request issued by a local upper layer helps the policy function that

resides out of the MIS to make optimal handover decisions for different applications when multiple links are available in an MN. However, a remotely initiated MIS_Link_Get_Parameters request from the network side enables the network to collect the status information on multiple links in an MN through the currently connected link.

7.4.12.2 MIS_Link_Get_Parameters.request

7.4.12.2.1 Function

This primitive is invoked by an MIS user to discover the status of the currently connected and potentially available links.

7.4.12.2.2 Semantics of the service primitive

MIS_Link_Get_Parameters.request (
DestinationIdentifier,
DeviceStatesRequest,
LinkIdentifierList,
GroupLinkIdentifier,
GetStatusRequestSet
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies the local MISF or a remote MISF that shall be the destination of this request.
DeviceStatesRequest	DEV_STATES_REQ	(Optional) List of device states being requested.
LinkIdentifierList	LIST(LINK_ID)	(Optional) List of link identifiers for which status is requested. If the list is empty, return the status of all available links. This parameter shall be used if and only if DestinationIdentifier is an MISF ID.
GroupLinkIdentifier	NET_TYPE_INC	(Optional) Identifier of a group of links for which status is requested. This parameter shall be used if and only if DestinationIdentifier is an MISF Group ID.
GetStatusRequestSet	LINK_STATUS_REQ	Indicate which link status(es) is being requested.

7.4.12.2.3 When generated

This primitive is invoked by an MIS user when it wants to request the status information of a set of local or remote links.

7.4.12.2.4 Effect of receipt

If the destination of the request is the local MISF itself, the local MISF gets the requested information on the status of the specified local links and responds with an MIS_Link_Get_Parameters.confirm. If the destination of the request is a remote MISF, the local MISF generates and sends an MIS_Link_Get_Parameters request message to the remote MISF.

7.4.12.3 MIS_Link_Get_Parameters.confirm

7.4.12.3.1 Function

This primitive is issued by an MISF to report the requested status of a set of specific local or remote links in response to an MIS_Link_Get_Parameters request from a local or remote MIS user.

7.4.12.3.2 Semantics of the service primitive

MIS_Link_Get_Parameters.confirm (
SourceIdentifier,
Status,
DeviceStatesResponseList,
GetStatusResponseList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
Status	STATUS	Status of operation.
DeviceStatesResponseList[a]	LIST(DEV_STATES_RSP)	(Optional) List of device states responses.
GetStatusResponseList[a]	LIST(SEQUENCE(LINK_ID, LINK_STATUS_RSP))	List of link status responses.

[a]This parameter is not included if Status does not indicate "Success."

7.4.12.3.3 When generated

This primitive returns the results of an MIS_Link_Get_Parameters request to the requesting MIS user.

7.4.12.3.4 Effect of receipt

Upon receipt of the link status information, the MIS user makes appropriate decisions and takes suitable actions. However, if Status does not indicate "Success," the recipient performs appropriate error handling.

7.4.13 MIS_Link_Configure_Thresholds

7.4.13.1 General

The MIS_Link_Configure_Thresholds is issued by an upper layer entity to configure parameter report thresholds of a lower layer. The destination of an MIS_Link_Configure_Thresholds command is local or remote. This command configures one or more thresholds on a link. When a given threshold is crossed, an MIS_Link_Parameters_Report notification shall be sent to all MIS users that are subscribed to this threshold-crossing event.

7.4.13.2 MIS_Link_Configure_Thresholds.request

7.4.13.2.1 Function

This primitive is issued by an MIS user to configure thresholds of a lower layer link.

7.4.13.2.2 Semantics of the service primitive

MIS_Link_Configure_Thresholds.request (
DestinationIdentifier,
ResponseFlag,
LinkIdentifier,
GroupLinkIdentifier,
ConfigureRequestList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies the local MISF or a remote MISF that shall be the destination of this request.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.
LinkIdentifier	LINK_TUPLE_ID	(Optional) Identifier of the link to be configured. This parameter shall be used if and only if DestinationIdentifier is an MISF ID.
GroupLinkIdentifier	NET_TYPE_INC	(Optional) Identifier of a group of links to be configured. This parameter shall be used if and only if DestinationIdentifier is an MISF Group ID.
ConfigureRequestList	LIST(LINK_CFG_PARAM)	A list of link threshold parameters.

[a]If the ResponseFlag parameter is not present, the MISF shall generate a request message, otherwise the MISF generates either a request or an indication message, based on the ResponseFlag parameter.

7.4.13.2.3 When generated

This primitive is invoked by an MIS user when it attempts to configure thresholds of a local or remote lower layer link.

7.4.13.2.4 Effect of receipt

If the destination of the request is the local MISF itself, the local MISF issues a Link_Configure_Thresholds.request to the lower layer link to set the thresholds for the link according to the specified configuration parameters.

If the destination of the request is a remote MISF, based on the ResponseFlag parameter, the local MISF generates and sends an MIS_Link_Configure_Thresholds request or an MIS Link_Configure_Thresholds indication message to the remote MISF. Upon the receipt of the message, the remote MISF then issues a Link_Configure_Thresholds request to the lower layer link to set the thresholds for the link according to the specified configuration parameters.

7.4.13.3 MIS_Link_Configure_Thresholds.confirm

7.4.13.3.1 Function

This primitive is issued by an MISF to report the result of an MIS_Link_Configure_Thresholds request.

7.4.13.3.2 Semantics of the service primitive

```
MIS_Link_Configure_Thresholds.confirm    (
                                         SourceIdentifier,
                                         Status,
                                         LinkIdentifier,
                                         ConfigureResponseList
                                         )
```

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
Status	STATUS	Status of operation.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link configured.
ConfigureResponseList[a]	LIST(LINK_CFG_STATUS)	A list of the configuration status for each requested link threshold parameter.

[a]This parameter is not included if Status does not indicate "Success."

7.4.13.3.3 When generated

This primitive returns the result of an MIS_Link_Configure_Thresholds request to the requesting MIS user.

7.4.13.3.4 Effect of receipt

Upon receipt of the result, the MIS user makes appropriate evaluations and takes any suitable actions. However, if Status does not indicate "Success," the recipient performs appropriate error handling.

7.4.14 MIS_Link_Actions

7.4.14.1 MIS_Link_Actions.request

7.4.14.1.1 Function

This primitive is used by an MIS user to control the behavior of a set of local or remote lower layer links.

7.4.14.1.2 Semantics of service primitive

MIS_Link_Actions.request (
Destination Identifier,
ResponseFlag,
LinkActionsList,
GroupLinkActionsList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies the local MISF or a remote MISF that shall be the destination of this request.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.
LinkActionsList	LIST(LINK_ACTION_REQ)	(Optional) Specifies the suggested actions. This parameter shall be used if and only if DestinationIdentifier is an MISF ID.
GroupLinkActionsList	LIST(MULTICAST_ACTION_REQ)	(Optional) Specifies the suggested actions for a group of links. This parameter shall be used if and only if DestinationIdentifier is an MISF Group ID.

[a]If the ResponseFlag parameter is not present, the MISF shall generate a request message, otherwise the MISF generates either a request or an indication message, based on the ResponseFlag parameter.

7.4.14.1.3 When generated

This primitive is invoked by an MIS user when it attempts to control the behavior of a set of local or remote lower layer links.

7.4.14.1.4 Effect on receipt

If the destination of the request is the local MISF itself, the local MISF issues a Link_Action.request(s) to the specified lower layer link(s).

If the destination of the request is a remote MISF, the local MISF generates and sends an MIS_Link_Actions request message to the remote MISF. Upon the receipt of the message, the remote MISF then issues Link_Action.request(s) to the specified lower layer link(s).

7.4.14.2 MIS_Link_Actions.confirm

7.4.14.2.1 Function

This primitive is issued by an MISF to report the result of an MIS_Link_Actions request.

7.4.14.2.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MIS_Link_Actions.confirm    (
                            SourceIdentifier,
                            Status,
                            LinkActionsResultList
                            )
```

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is either the local MISF or a remote MISF.
Status	STATUS	Status of operation.
LinkActionsResultList[a]	LIST(LINK_ACTION_RSP)	Contain the result of the request link actions.

[a]This parameter is not included if Status does not indicate “Success.”

7.4.14.2.3 When generated

This primitive returns the result of an MIS_Link_Actions.request to the requesting MIS user.

7.4.14.2.4 Effect on receipt

Upon receipt of the result, the MIS user makes appropriate evaluations and takes any suitable actions. However, if Status does not indicate “Success,” the recipient performs appropriate error handling.

7.4.15 MIS_Get_Information

7.4.15.1 MIS_Get_Information.request

7.4.15.1.1 Function

This primitive is used by an MIS user to request information from an MIS information server. The information query is related to a specific interface, attributes to the network interface, as well as the entire network capability. The service primitive has the flexibility to query either a specific data within a network interface or extended schema of a given network. It is assumed that the available information could be broadcast in access-technology-specific manner such as in IEEE Std 802.11-2012 and IEEE Std 802.16-2012.

7.4.15.1.2 Semantics of service primitive

MIS_Get_Information.request (
DestinationIdentifier,
InfoQueryBinaryDataList,
InfoQueryRDFDataList,
InfoQueryRDFSchemaURL,
InfoQueryRDFSchemaList,
MaxResponseSize,
QuerierNetworkType,
UnauthenticatedInformationRequest
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	The local MISF or a remote MISF that shall be the destination of this request.
InfoQueryBinaryDataList	LIST(IQ_BIN_DATA)	(Optional) A list of TLV queries. The order of the queries in the list identifies the priority of the query. The first query has the highest priority to be processed by MIIS. See Table E.15 for detailed definition.
InfoQueryRDFDataList	LIST(IQ_RDF_DATA)	(Optional) A list of RDF queries. The order of the queries in the list identifies the priority of the query. The first query has the highest priority to be processed by MIIS. See Table E.16 for detailed definition.
InfoQueryRDFSchemaURL	BOOLEAN	(Optional) A RDF Schema URL query. This field is required only when the value is “TRUE,” which indicates to query a list of RDF schema URLs.
InfoQueryRDFSchemaList	LIST(IQ_RDF_SCHM)	(Optional) A list of RDF schema queries. The order of the queries in the list identifies the priority of the query. The first query has the highest priority to be processed by MIIS.
MaxResponseSize	UNSIGNED_INT(2)	(Optional) This field specifies the maximum size of Info Response parameters in MIS_Get_Information response primitive in octets. If this field is not specified, the maximum size is set to 65535. The actual maximum size forced by the IS server can be smaller than that specified by the IS client.
QuerierNetworkType	NETWORK_TYPE	(Optional) The type of the network being used by the querier. This parameter is valid only with InfoQuery-BinaryDataList and InfoQueryRDFDataList.
UnauthenticatedInformation Request	BOOLEAN	The value of unauthenticated information request (UIR) bit to be set in the MIS_Get_Information request message sent to the remote MISF.

One and only one of the following parameters is specified:

— InfoQueryBinaryDataList
— InfoQueryRDFDataList
— InfoQueryRDFSchemaURL
— InfoQueryRDFSchemaList

7.4.15.1.3 When generated

This primitive is generated by an MIS user that is seeking to retrieve information.

The order of the queries in each of InfoQueryBinaryDataList, InfoQueryRDFDataList, and InfoQueryRDFSchemaList parameters identifies the priority of the query. The first query has the highest priority to be processed by MIIS.

7.4.15.1.4 Effect on receipt

If the DestinationIdentifier contains a remote MISF, then the recipient shall forward the query in an MIS_Get_Information request message to the designated MIIS server. If the DestinationIdentifier is for the local MISF, then the recipient shall interpret the query request and retrieve the specified information.

7.4.15.2 MIS_Get_Information.indication

7.4.15.2.1 Function

This primitive is used by the MISF to indicate that an MIS_Get_Information request message is received from a peer MISF.

7.4.15.2.2 Semantics of service primitive

```
MIS_Get_Information.indication   (
                                 SourceIdentifier,
                                 InfoQueryBinaryDataList,
                                 InfoQueryRDFDataList,
                                 InfoQueryRDFSchemaURL,
                                 InfoQueryRDFSchemaList,
                                 MaxResponseSize,
                                 QuerierNetworkType,
                                 UnathenticatedInformationRequest
                                 )
```

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies the MISF ID of the node that sent the MIS_Get_Information request message.
InfoQueryBinaryDataList	LIST(IQ_BIN_DATA)	(Optional) A list of TLV queries. The order of the queries in the list identifies the priority of the query. The first query has the highest priority to be processed by MIIS. See Table E.15 for detailed definition.
InfoQueryRDFDataList	LIST(IQ_RDF_DATA)	(Optional) A list of RDF queries. The order of the queries in the list identifies the priority of the query. The first query has the highest priority to be processed by MIIS.
InfoQueryRDFSchemaURL	BOOLEAN	(Optional) A RDF Schema URL query. This field is required only when the value is "TRUE," which indicates to query a list of RDF schema URLs.
InfoQueryRDFSchemaList	LIST(IQ_RDF_SCHM)	(Optional) A list of RDF schema queries. The order of the queries in the list identifies the priority of the query. The first query has the highest priority to be processed by MIIS.
MaxResponseSize	UNSIGNED_INT(2)	(Optional) This field specifies the maximum size of Info Response parameters in MIS_Get_Information response primitive in octets. If this field is not specified, the maximum size is set to 65 535. The actual maximum size forced by the IS server can be smaller than that specified by the IS client.
QuerierNetworkType	NETWORK_TYPE	(Optional) The type of the network being used by the querier. This parameter is valid only with InfoQueryBinaryDataList and InfoQueryRDFDataList.
UnauthenticatedInformationRequest	BOOLEAN	The value of UIR bit contained in the MIS_Get_Information request message received from the remote MISF.

7.4.15.2.3 When generated

This primitive is generated by the MISF on receiving an MIS_Get_Information request message from a peer MISF. The order of the queries in each of InfoQueryBinaryDataList, InfoQueryRDFDataList, and InfoQueryRDFSchemaList parameters identifies the priority of the query. The first query has the highest priority to be processed by MIIS. Thus the order of the queries is maintained as indicated by the request message.

7.4.15.2.4 Effect on receipt

The recipient interprets the query request and retrieves the specified information. Once the information is retrieved, the recipient replies with the MIS_Get_Information.response primitive.

7.4.15.3 MIS_Get_Information.response

7.4.15.3.1 Function

This primitive is used by an MIS user (i.e., MIIS Server) to respond to an MIS_Get_Information.indication primitive.

7.4.15.3.2 Semantics of service primitive

MIS_Get_Information.response (
DestinationIdentifier,
Status,
InfoResponseBinaryDataList,
InfoResponseRDFDataList,
InfoResponseRDFSchemaURLList,
InfoResponseRDFSchemaList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	The local MISF or a remote MISF that shall be the destination of this response.
Status	STATUS	Status of operation. The response lists contains meaningful data if and only if the status is '0'.
InfoResponseBinaryDataList	LIST(IR_BIN_DATA)	(Optional) A list of TLV query responses. The list shall be sorted from most preferred first to least preferred last.
InfoResponseRDFDataList	LIST(IR_RDF_DATA)	(Optional) A list of RDF query responses. The list shall be sorted from most preferred first to least preferred last.
InfoResponseRDFSchemaURLList	LIST(IR_SCHM_URL)	(Optional) A list of RDF Schema URL. The list shall be sorted from most preferred first to least preferred last.
InfoResponseRDFSchemaList	LIST(IR_RDF_SCHM)	(Optional) A list of RDF schema query responses. The list shall be sorted from most preferred first to least preferred last.

7.4.15.3.3 When generated

This primitive is generated by an MIS user in response to a received MIS_Get_Information.indication primitive. When the size of the Info Response parameters exceeds the maximum size specified in the MaxResponseSize parameter from MIS_Get_Information.indication primitive, one or more of the lower order list elements in Info Response parameters shall be omitted.

7.4.15.3.4 Effect on receipt

The recipient returns an MIS_Get_Information response message to the designated MIIS client.

7.4.15.4 MIS_Get_Information.confirm

7.4.15.4.1 Function

This primitive is used by the MISF to respond to an MIS_Get_Information.request primitive.

7.4.15.4.2 Semantics of service primitive

MIS_Get_Information.confirm (
SourceIdentifier,
Status,
InfoResponseBinaryDataList,
InfoResponseRDFDataList,
InfoResponseRDFSchemaURLList,
InfoResponseRDFSchemaList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies the MISF ID of the node that invoked MIS_Get_Information.response.
Status	STATUS	Status of operation. The response lists contains meaningful data if and only if the status is '0'.
InfoResponseBinaryDataList	LIST(IR_BIN_DATA)	(Optional) A list of TLV query responses. The list shall be sorted from most preferred first to least preferred last.
InfoResponseRDFDataList	LIST(IR_RDF_DATA)	(Optional) A list of RDF query responses. The list shall be sorted from most preferred first to least preferred last.
InfoResponseRDFSchemaURLList	LIST(IR_SCHM_URL)	(Optional) A list of RDF Schema URL. The list shall be sorted from most preferred first to least preferred last.
InfoResponseRDFSchemaList	LIST(IR_RDF_SCHM)	(Optional) A list of RDF schema query responses. The list shall be sorted from most preferred first to least preferred last.

7.4.15.4.3 When generated

This primitive is generated by the MISF on receiving an MIS_Get_Information Response message from a peer MISF.

7.4.15.4.4 Effect on receipt

The MIS user that requested the information utilizes the Info Response parameters and takes suitable action. However, if Status does not indicate "Success," the recipient ignores any other returned values and, instead, performs appropriate error handling.

When the size of the Info Response parameters exceeds the maximum size specified in the MaxResponseSize parameter from MIS_Get_Information.request primitive, one or more of the lower order list elements in Info Response parameters shall be omitted.

7.4.16 MIS_Push_Information

7.4.16.1 MIS_Push_Information.request

7.4.16.1.1 Function

This primitive is used by an MIS user (i.e., MIIS Server) to push information to the MN. MIS_Push_Information is generated by the MIIS Server to update policy information following a successful registration. This primitive is generated at any time during the life time of the registration.

7.4.16.1.2 Semantics of service primitive

MIS_Push_Information.request (
DestinationIdentifier,
InfoResponseBinaryDataList,
InfoResponseRDFDataList,
InfoResponseRDFSchemaURLList,
InfoResponseRDFSchemaList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	The remote MISF, which shall be the destination of this request.
InfoResponseBinaryDataList	LIST(IR_BIN_DATA)	(Optional) A list of binary representations of Information Elements. This list shall be sorted from most preferred first to least preferred last. This list includes vendor-specific IEs for representing network policies.
InfoResponseRDFDataList	LIST(IR_RDF_DATA)	(Optional) A list of network information in RDF. The list shall be sorted from most preferred first to least preferred last. This list includes operator-specific RDF data that is used to represent operator policies.
InfoResponseRDFSchemaURLList	LIST(IR_SCHM_URL)	(Optional) A list of RDF Schema URL supported by the network. The list shall be sorted from most preferred first to least preferred last.
InfoResponseRDFSchemaList	LIST(IR_RDF_SCHM)	(Optional) A list of RDF Schema content supported by the network. The list shall be sorted from most preferred first to least preferred last.

At least one of the following parameters is specified:

— InfoResponseBinaryDataList
— InfoResponseRDFDataList
— InfoResponseRDFSchemaURLList
— InfoResponseRDFSchemaList

7.4.16.1.3 When generated

This primitive is generated by the MIIS server to update any policies on the MN or to update any other IEs from the MIIS on to the MN.

7.4.16.1.4 Effect on receipt

The recipient returns an MIS_Push_Information indication message.

7.4.16.2 MIS_Push_Information.indication

7.4.16.2.1 Function

This primitive is used by the MISF to notify MIS users of the information. This primitive is the result of receipt of an MIS_Push_information indication message from a remote MISF.

7.4.16.2.2 Semantics of service primitive

MIS_Push_Information.indication (
SourceIdentifier,
InfoResponseBinaryDataList,
InfoResponseRDFDataList,
InfoResponseRDFSchemaURLList,
InfoResponseRDFSchemaList
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	The remote MISF, which is the source of this request.
InfoResponseBinaryDataList	LIST(IR_BIN_DATA)	(Optional) A list of binary representations of Information Elements. This list shall be sorted from most preferred first to least preferred last. This list includes vendor-specific IEs for representing network policies.
InfoResponseRDFDataList	LIST(IR_RDF_DATA)	(Optional) A list of network information in RDF. The list shall be sorted from most preferred first to least preferred last. This list includes operator-specific RDF data that is used to represent operator policies.
InfoResponseRDFSchemaURLList	LIST(IR_SCHM_URL)	(Optional) A list of RDF Schema URL supported by the network. The list shall be sorted from most preferred first to least preferred last.
InfoResponseRDFSchemaList	LIST(IR_RDF_SCHM)	(Optional) A list of RDF Schema content supported by the network. The list shall be sorted from most preferred first to least preferred last.

7.4.16.2.3 When generated

This primitive is generated by the MISF upon receiving an MIS_Push_Information indication message.

7.4.16.2.4 Effect on receipt

Upper layer entities take different actions upon notification.

7.4.17 MIS_Push_Key

7.4.17.1 MIS_Push_key.request

7.4.17.1.1 Function

This primitive is used to request a remote MISF (PoS to install a key[s] in a target PoA[s] or PoS[s]).

7.4.17.1.2 Semantics of service primitive

MIS_Push_key.request (
DestinationIdentifier,
LinkTupleIdentifierList
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF that shall be the destination of this request.
LinkTupleIdentifierList	LIST(LINK_TUPLE_ID)	This identifies a list of links of target PoAs for which keys are pushed.

7.4.17.1.3 When generated

This primitive is generated by an MIS user in the MN to request a remote MISF in the serving PoS to install a key in a target PoA.

7.4.17.1.4 Effect on receipt

The local MISF shall generate an MIS_Push_Key request message to the remote MISF.

7.4.17.2 MIS_Push_key.indication

7.4.17.2.1 Function

This primitive is used to pass a key to the corresponding MIS user on the serving PoS.

7.4.17.2.2 Semantics of service primitive

MIS_Push_key.indication (
SourceIdentifier,
KeyMapping
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker, which is a remote MISF.
KeyMapping	KEY_MAPPING	This specifies a mapping of a link identifier for which the key is pushed and a lifetime.

7.4.17.2.3 When generated

This primitive is generated by the local MISF after receiving an MIS_Push_Key request message from the remote MISF.

7.4.17.2.4 Effect on receipt

A media-specific key is delivered to the corresponding MIS user.

7.4.17.3 MIS_Push_key.response

7.4.17.3.1 Function

This primitive is used to indicate that the key installation request has been received and MIS user has executed it.

7.4.17.3.2 Semantics of service primitive

MIS_Push_key.response (
DestinationIdentifier,
LinkTupleIdentifierList,
Status
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF that shall be the destination of this response.
LinkTupleIdentifierList	LIST(LINK_TUPLE_ID)	This identifies a list of links for which keys are pushed.
Status	STATUS	This represents the operation result.

7.4.17.3.3 When generated

This primitive is generated by an MIS user after receiving an MIS_Push_Key.indication primitive.

7.4.17.3.4 Effect on receipt

The local MISF shall generate an MIS_Push_Key response message to the remote MISF.

7.4.17.4 MIS_Push_Key.confirm

7.4.17.4.1 Function

This primitive is used to notify the MIS user (in MN side) about the status of the requested operation.

7.4.17.4.2 Semantics of service primitive

MIS_Push_Key.confirm (
SourceIdentifier,
LinkTupleIdentifierList,
Status
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker, which is a remote MISF.
LinkTupleIdentifierList	LIST(LINK_TUPLE_ID)	This identifies a list of links for which keys are pushed.
Status	STATUS	This represents the operation result.

7.4.17.4.3 When generated

This primitive is generated after receiving an MIS_Push_Key response message.

7.4.17.4.4 Effect on receipt

A media-specific key shall be installed in the link layer.

7.4.18 MIS_LL_Auth

The primitives defined are to carry out a proactive authentication over MIS between the MN and the PoS using link-layer frames. The authentication is conducted with the media-specific authenticator that serves the target PoA.

7.4.18.1 MIS_LL_Auth.request

7.4.18.1.1 Function

This primitive carries link-layer frames for authentication purposes.

7.4.18.1.2 Semantics of service primitive

MIS_LL_Auth.request (
DestinationIdentifier,
LinkIdentifier,
LLInformation
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF that shall be the destination of this request.
LinkIdentifier	LINK_TUPLE_ID	This identifies a PoA that is also the authenticator.
LLInformation	LL_FRAMES	This carries link-layer frames.

7.4.18.1.3 When generated

This primitive is generated by an MIS user to start an authentication process based on link-layer frames.

7.4.18.1.4 Effect on receipt

The local MISF shall generate an MIS_LL_Auth request message to the remote MISF.

7.4.18.2 MIS_LL_Auth.indication

7.4.18.2.1 Function

This primitive is used by the remote MISF to notify the corresponding MIS user about the reception of an MIS_LL_Auth request message.

7.4.18.2.2 Semantics of service primitive

MIS_LL_Auth.indication (
SourceIdentifier,
LinkIdentifier,
LLInformation
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker, which is a remote MISF.
LinkIdentifier	LINK_TUPLE_ID	This identifies a PoA that is also the authenticator.
LLInformation	LL_FRAMES	This carries link-layer frames.

7.4.18.2.3 When generated

This primitive is generated by a remote MISF after receiving an MIS_LL_Auth request message.

7.4.18.2.4 Effect on receipt

The MIS user shall generate an MIS_LL_Auth.response primitive.

7.4.18.3 MIS_LL_Auth.response

7.4.18.3.1 Function

This primitive is used by an MIS user to provide the link-layer frames to the local MISF.

7.4.18.3.2 Semantics of service primitive

MIS_LL_Auth.response (
DestinationIdentifier,
LinkIdentifier,
LLInformation,
Status
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF that shall be the destination of this response.
LinkIdentifier	LINK_TUPLE_ID	This identifies a PoA that is also the authenticator.
LLInformation	LL_FRAMES	This carries link-layer frames.
Status	STATUS	Status of the authentication.

7.4.18.3.3 When generated

This primitive is generated after receiving an MIS_LL_Auth.indication primitive.

7.4.18.3.4 Effect on receipt

The local MISF shall generate an MIS_LL_Auth response message in order to provide the required information until the authentication is finished.

7.4.18.4 MIS_LL_Auth.confirm

7.4.18.4.1 Function

This primitive is used to notify the corresponding MIS user about the reception of an MIS_LL_Auth response message.

7.4.18.4.2 Semantics of service primitive

MIS_LL_Auth.confirm (
SourceIdentifier,
LLInformation,
Status
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker, which is a remote MISF.
LLInformation	LL_FRAMES	This carries link-layer frames.
Status	STATUS	Status of the authentication.

7.4.18.4.3 When generated

This primitive is generated by the remote MISF after receiving an MIS_LL_Auth response message.

7.4.18.4.4 Effect on receipt

The MIS user may generate an MIS_LL_Auth.request primitive unless the authentication is completed.

7.4.19 MIS_Configuration_Update

7.4.19.1 MIS_Configuration_Update.request

7.4.19.1.1 Function

This primitive is generated by a PoS to update the configuration of one or more MN(s) or other PoS(s).

7.4.19.1.2 Semantics of service primitive

MIS_Configuration_Update.request (
DestinationIdentifier,
ConfigurationData,
ResponseFlag
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies MISF ID of the remote MISF(s) to be configured.
ConfigurationData	OCTET_STRING	Configuration data. Examples of this parameter include firmware and management parameters.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.

[a] If the ResponseFlag parameter is not present, the MISF shall generate a request message, otherwise the MISF generates either a request or an indication message, based on the ResponseFlag parameter.

7.4.19.1.3 When generated

Upon receipt of this primitive an MISF shall send an MIS_Configuration_Update request or indication message to the destination, based on the ResponseFlag parameter.

7.4.19.1.4 Effect on receipt

Upon receipt of this primitive, MISF on the PoS sends the corresponding MIS_Configuration_Update indication message or MIS_Configuration_Update request message to the MN(s) or other PoS(s).

7.4.19.2 MIS_Configuration_Update.indication

7.4.19.2.1 Function

This primitive is generated by an MISF to update the configuration of one or more MN(s) or other PoS(s).

7.4.19.2.2 Semantics of service primitive

MIS_Configuration_Update.indication (
SourceIdentifier,
DestinationIdentifier,
ConfigurationData,
ResponseFlag
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies MISF ID of the remote MISF that sent either the MIS_Configuration_Update indication message or the MIS_Configuration_Update request message.
DestinationIdentifier	MISF_ID	The target MISF identifier for the operation.
ConfigurationData	OCTET_STRING	Configuration data. Examples of this parameter include firmware and management parameters.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.

[a] If the ResponseFlag parameter is not present, the MISF shall generate a response message, otherwise the MISF may generate either a response message or no response message, based on the ResponseFlag parameter.

7.4.19.2.3 When generated

This primitive is generated by an MISF on an MN or a PoS when receiving an MIS_Configuration_Update indication message or MIS_Configuration_Update request message from a remote MISF.

7.4.19.2.4 Effect on receipt

Upon receipt of this primitive, an MIS user on an MN or a PoS should modify its configuration using the ConfigurationData parameter. If the ResponseFlag parameter is not present, or ResponseFlag parameter is present and its value is '1', the MIS user shall generate an MIS_Configuration_Update.response primitive.

7.4.19.3 MIS_Configuration_Update.response

7.4.19.3.1 Function

This primitive is generated by an MIS user to acknowledge the result of an MIS_Configuration_Update request message from a PoS.

7.4.19.3.2 Semantics of service primitive

MIS_Configuration_Update.response (
DestinationIdentifier,
Status
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies the requestor of the configuration update.
Status	STATUS	Status of operation.

7.4.19.3.3 When generated

An MIS user generates this primitive after receiving and processing an MIS_Configuration_Update.indication primitive.

7.4.19.3.4 Effect on receipt

The status of the configuration update operation is noted.

7.4.19.4 MIS_Configuration_Update.confirm

7.4.19.4.1 Function

This primitive is generated by an MISF that receives an MIS_Configuration_Update response to indicate the status of the configuration update.

7.4.19.4.2 Semantics of service primitive

MIS_Configuration_Update.confirm (
SourceIdentifier,
Status
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies the responder of the configuration update.
Status	STATUS	Status of operation.

7.4.19.4.3 When generated

An MISF generates this primitive after receiving and processing an MIS_Configuration_Update response message.

7.4.19.4.4 Effect on receipt

The status of the configuration update operation is noted.

7.4.20 MIS_Pull_Group_Manipulate

7.4.20.1 MIS_Pull_Group_Manipulate.request

7.4.20.1.1 Function

This primitive is generated by an MN or a PoS to manipulate its own group membership.

7.4.20.1.2 Semantics of service primitive

MIS_Pull_Group_Manipulate.request (
DestinationIdentifier,
TargetIdentifier,
GroupAction
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies MISF ID of the remote MISF peers. DestinationIdentifier may be different from TargetIdentifier.
TargetIdentifier	MISF_ID	The target MISF group identifier for the group operation.
GroupAction	GROUP_MGT_ACTION	The action to be taken: Join or leave the group.

7.4.20.1.3 When generated

The MIS user generates this primitive to request joining or leaving a group.

7.4.20.1.4 Effect on receipt

Upon receipt of this primitive, MISF on the MN or PoS sends the corresponding MIS_Pull_Group_Manipulate request message to the PoS.

7.4.20.2 MIS_Pull_Group_Manipulate.indication

7.4.20.2.1 Function

This primitive is used by an MISF to notify an MIS user that an MIS_Pull_Group_Manipulate request message has been received.

7.4.20.2.2 Semantics of service primitive

MIS_Pull_Group_Manipulate.indication (
SourceIdentifier,
TargetIdentifier,
GroupAction
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies MISF ID of the remote MISF that issued MIS_Pull_Group_Manipulate.request.
TargetIdentifier	MISF_ID	The target MISF group identifier for the group operation.
GroupAction	GROUP_MGT_ACTION	The action to be taken: Join or leave the group.

7.4.20.2.3 When generated

This primitive is generated by an MISF on a PoS when receiving an MIS_Pull_Group_Manipulate request message from a remote MISF.

7.4.20.2.4 Effect on receipt

Upon receipt of this primitive, an MIS user on a PoS shall take the required actions as the action specified in GroupAction.

7.4.20.3 MIS_Pull_Group_Manipulate.response

7.4.20.3.1 Function

This primitive is generated by an MIS user in a PoS with group manager to acknowledge result of an MIS_Pull_Group_Manipulate request from an MN or a PoS.

7.4.20.3.2 Semantics of service primitive

MIS_Pull_Group_Manipulate.response (
DestinationIdentifier,
TargetIdentifier,
TransportAddress,
MasterGroupKey,
SubgroupRange,
UserSpecificData,
CompleteSubtree,
ComplementSubtreeFlag,
GroupKeyData,
VerifyGroupCode,
GroupStatus
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies the MISF ID of the destination of the primitive.
TargetIdentifier	MISF_ID	The target MISF group identifier for the group operation.
TransportAddress	TRANSPORT_ADDR	(Optional) Multicast or unicast address corresponding with the target group identifier. A unicast address is used for a two-member group.
MasterGroupKey	MGK	(Optional) The master group key associated with the target MISF group identifier.
SubgroupRange[a]	SUBGROUP_RANGE	(Optional) Subgroup to process the command.
UserSpecificData[b]	OCTET_STRING	(Optional) Auxiliary data.
CompleteSubtree[c]	COMPLETE_SUBTREE	(Optional) Complete Subtree data.
ComplementSubtreeFlag	SUBTREE_FLAG	(Optional) Flag to interpret the complete Subtree data. (See 9.5.2.)
GroupKeyData[d]	GROUP_KEY_DATA	(Optional) Encrypted group key.
VerifyGroupCode	VERIFY_GROUP_KEY	(Optional) Verification data for group key.
GroupStatus	GROUP_STATUS	Status of the group operation.

[a] SubgroupRange parameter shall be present for a fragmented group key block (GKB).
[b] The UserSpecificData parameter is used to convey additional information such as version information of the GKB used or additional credentials.
[c] If CompleteSubtree is a list of length '0', no Node Index is matched in the CompleteSubtree.
[d] In case the GroupKeyData parameter is not present, the CompleteSubtree parameter shall be present.

7.4.20.3.3 When generated

An MIS user at the PoS with group manager generates this primitive after receiving and processing of MIS_Pull_Group_Manipulate request message. This primitive returns the status of the action asked in the request. Optionally, it responds with the security mechanisms required by the group.

7.4.20.3.4 Effect on receipt

MIS_Pull_Group_Manipulate response message is sent back to the requester.

7.4.20.4 MIS_Pull_Group_Manipulate.confirm

7.4.20.4.1 Function

This primitive is generated by an MISF that receives an MIS_Pull_Group_Manipulate response to indicate the status of the group manipulation. The status of the group manipulation provides information regarding the result of a group join or leave operation, indicating the status after the command execution.

7.4.20.4.2 Semantics of service primitive

```
MIS_Pull_Group_Manipulate.confirm    (
                                     SourceIdentifier,
                                     TargetIdentifier,
                                     GroupStatus
                                     )
```

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies the MISF ID of the remote MISF.
TargetIdentifier	MISF_ID	The target MISF group identifier for the group operation.
GroupStatus	GROUP_STATUS	Status of the group operation.

7.4.20.4.3 When generated

This primitive is sent to the MIS user after the MISF receives an MIS_Pull_Group_Manipulate response message.

7.4.20.4.4 Effect on receipt

The status of the group operation is noted.

7.4.21 MIS_Push_Group_Manipulate

7.4.21.1 MIS_Push_Group_Manipulate.request

7.4.21.1.1 Function

This primitive is generated by the MIS user of a PoS with group manager to manipulate group membership of one or more MN(s) or other PoS(s).

7.4.21.1.2 Semantics of service primitive

MIS_Push_Group_Manipulate.request (
DestinationIdentifier,
ResponseFlag,
GroupKeyUpdateFlag,
TargetIdentifier,
TransportAddress,
MasterGroupKey,
SubgroupRange,
UserSpecificData,
CompleteSubtree,
ComplementSubtreeFlag,
GroupKeyData,
VerifyGroupCode
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies either MISF ID of the specific MISF peer or MISF Group ID of the remote MISF peers. DestinationIdentifier may be different from TargetIdentifier.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.
GroupKeyUpdateFlag	GROUP_KEY_UPDATE_FLAG	Flag that represents whether or not a group key in GroupKeyData is updated.
TargetIdentifier	MISF_ID	The target MISF group identifier for the group operation.
TransportAddress	TRANSPORT_ADDR	(Optional) Multicast or unicast address corresponding with the target group identifier. A unicast address is used for a two-member group.
MasterGroupKey	MGK	(Optional) The master group key associated with the target MISF group identifier.
SubgroupRange[b]	SUBGROUP_RANGE	(Optional) Subgroup to process the command.
UserSpecificData[c]	OCTET_STRING	(Optional) Auxiliary data.
CompleteSubtree[d]	COMPLETE_SUBTREE	(Optional) Complete Subtree data.
ComplementSubtreeFlag	SUBTREE_FLAG	(Optional) Flag to interpret the complete Subtree data. (See 9.5.2.)
GroupKeyData[e]	GROUP_KEY_DATA	(Optional) Encrypted group key.
VerifyGroupCode	VERIFY_GROUP_KEY	(Optional) Verification data for group key.

[a] If the ResponseFlag parameter is not present, the MISF shall generate a request message, otherwise the MISF generates either a request or an indication message, based on the ResponseFlag parameter.
[b] SubgroupRange parameter shall be present for a fragmented GKB.
[c] The UserSpecificData parameter is used to convey additional information such as version information of the GKB used or additional credentials.
[d] If CompleteSubtree is a list of length '0', no Node Index is matched in the CompleteSubtree. In case the CompleteSubtree is not present, the GroupKeyData parameter shall be present.
[e] In case the GroupKeyData parameter is not present, the CompleteSubtree parameter shall be present.

7.4.21.1.3 When generated

The MIS user generates this primitive to create, delete, or modify group membership.

7.4.21.1.4 Effect on receipt

Upon receipt of this primitive, MISF on the PoS sends the corresponding MIS_Push_Group_Manipulate indication message or MIS_Push_Group_Manipulate request message to the MN(s) or other PoS(s). The ResponseFlag TLV indicates which message shall be sent.

7.4.21.2 MIS_Push_Group_Manipulate.indication

7.4.21.2.1 Function

This primitive is used by an MISF to notify an MIS user that an MIS_Push_Group_Manipulate indication message or an MIS_Push_Group_Manipulate request message has been received.

7.4.21.2.2 Semantics of service primitive

MIS_Push_Group_Manipulate.indication (
SourceIdentifier,
ResponseFlag,
TargetIdentifier,
TransportAddress,
UserSpecificData,
GroupStatus
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies MISF ID of the remote MISF that issued either MIS_Push_Group_Manipulate indication message or MIS_Push_Group_Manipulate request message.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.
TargetIdentifier	MISF_ID	The target MISF group identifier for the group operation.
TransportAddress	TRANSPORT_ADDR	(Optional) Multicast or unicast address corresponding with the target group. A unicast address is used for a two-member group.
UserSpecificData	OCTET_STRING	(Optional) Auxiliary data.
GroupStatus	GROUP_STATUS	Status of the group.

[a] If the ResponseFlag parameter is not present, the MISF shall generate a response message, otherwise the MISF generates either a response message or no response message, based on the ResponseFlag parameter.

7.4.21.2.3 When generated

This primitive is generated by an MISF on an MN or a PoS when receiving an MIS_Push_Group_Manipulate indication message or an MIS_Push_Group_Manipulate request message from a remote MISF.

7.4.21.2.4 Effect on receipt

Upon reception of this primitive, an MIS user on an MN or a PoS should join or leave the group specified in the TargetIdentifier parameter. The MISF shall also de-capsulate the GroupKeyData, and install a group key derived by the decapsulation. The detailed procedure is described in 9.5.3.1.2. If the ResponseFlag parameter is not present, or ResponseFlag parameter is present and its value is ‘1’, the MIS user shall generate an MIS_Push_Group_Manipulate.response primitive.

7.4.21.3 MIS_Push_Group_Manipulate.response

7.4.21.3.1 Function

This primitive is generated by an MIS user to acknowledge the result of an MIS_Push_Group_Manipulate request from a PoS.

7.4.21.3.2 Semantics of service primitive

MIS_Push_Group_Manipulate.response (
DestinationIdentifier,
TargetIdentifier,
GroupStatus
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies the requestor of the group manipulation.
TargetIdentifier	MISF_ID	The target MISF group identifier for the group operation.
GroupStatus	GROUP_STATUS	Status of the group.

7.4.21.3.3 When generated

An MIS user generates this primitive after receiving and processing of MIS_Push_Group_Manipulate.indication primitive.

7.4.21.3.4 Effect on receipt

MIS_Push_Group_Manipulate response message is sent back to the group manipulate requester.

7.4.21.4 MIS_Push_Group_Manipulate.confirm

7.4.21.4.1 Function

This primitive is generated by an MISF that receives an MIS_Push_Group_Manipulate response to indicate the status of the group manipulation.

7.4.21.4.2 Semantics of service primitive

MIS_Push_Group_Manipulate.confirm (
SourceIdentifier,
TargetIdentifier,
GroupStatus
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies the responder of the group manipulation.
TargetIdentifier	MISF_ID	The target MISF group identifier for the group operation.
GroupStatus	GROUP_STATUS	Status of the group.

7.4.21.4.3 When generated

An MISF generates this primitive after receiving and processing an MIS_Push_Group_Manipulate request message.

7.4.21.4.4 Effect on receipt

The status of the group operation is noted.

7.4.22 MIS_Pull_Certificate

7.4.22.1 MIS_Pull_Certificate.request

7.4.22.1.1 Function

This primitive is generated by an MN or a PoS and it is used to request sending of a PoS certificate from the destination PoS to the requestor.

7.4.22.1.2 Semantics of service primitive

MIS_Pull_Certificate.request (
DestinationIdentifier
)

Parameter:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF that shall be the destination of this request.

7.4.22.1.3 When generated

An MN or a PoS generates this primitive for requesting a certificate of the PoS identified by DestinationIdentifier.

7.4.22.1.4 Effect on receipt

Upon receipt of this primitive, the MISF on the MN or PoS sends the corresponding MIS_Pull_Certificate request message to the destination PoS.

7.4.22.2 MIS_Pull_Certificate.indication

7.4.22.2.1 Function

This primitive is generated by an MISF that receives an MIS_Pull_Certificate request message in order to inform the MIS user.

7.4.22.2.2 Semantics of service primitive

MIS_Pull_Certificate.indication (
SourceIdentifier
)

Parameter:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is a remote MISF.

7.4.22.2.3 When generated

This primitive is generated by an MISF when an MIS_Pull_Certificate request message is received.

7.4.22.2.4 Effect on receipt

Upon reception of this primitive, the MIS user generates an MIS_Pull_Certificate.response to deliver a PoS certificate to the requester.

7.4.22.3 MIS_Pull_Certificate.response

7.4.22.3.1 Function

This primitive is generated by an MIS user in order to deliver a PoS certificate to an MN or other PoS.

7.4.22.3.2 Semantics of service primitive

MIS_Pull_Certificate.response (
DestinationIdentifier,
Certificate
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	This identifies a remote MISF that shall be the destination of this response.
Certificate	CERTIFICATE	A PoS's X.509 certificate for signature-based MIS protection.

7.4.22.3.3 When generated

An MIS user generates this primitive in response to MIS_Pull_Certificate.indication primitive.

7.4.22.3.4 Effect on receipt

Upon receipt of this primitive, the MISF on the PoS generates an MIS_Pull_Certificate response message to the destination MN or PoS.

7.4.22.4 MIS_Pull_Certificate.confirm

7.4.22.4.1 Function

This primitive is generated by an MISF that receives an MIS_Pull_Certificate response, in order to inform of the PoS certificate received by the MISF.

7.4.22.4.2 Semantics of service primitive

MIS_Pull_Certificate.confirm (
SourceIdentifier,
Certificate
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	This identifies the invoker of this primitive, which is a remote MISF.
Certificate	CERTIFICATE	A PoS's X.509 certificate for signature-based MIS protection as described in 8.4.2.

7.4.22.4.3 When generated

The MISF that receives an MIS_Pull_Certificate response message generates this primitive to indicate the PoS certificate.

7.4.22.4.4 Effect on receipt

Upon reception of this primitive, the MIS user installs the PoS certificate for signature verification of MIS PDUs sent by the remote MISF.

7.4.23 MIS_Push_Certificate

7.4.23.1 MIS_Push_Certificate.request

7.4.23.1.1 Function

This primitive is generated by an MIS user at the PoS to send a Certificate to a destination PoS(s) or MN(s).

7.4.23.1.2 Semantics of service primitive

MIS_Push_Certificate.request (
DestinationIdentifier,
ResponseFlag,
Certificate
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies the recipient(s) of the credential.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.
Certificate	CERTIFICATE	A PoS's X.509 certificate.

[a] If the ResponseFlag parameter is not present, the MISF shall generate a request message, otherwise the MISF generates either a request or an indication message, based on the ResponseFlag parameter.

7.4.23.1.3 When generated

A PoS generates this primitive for initial provisioning of credentials or for credential updates.

7.4.23.1.4 Effect on receipt

Upon receipt of this primitive, the MISF on the PoS sends the corresponding MIS_Push_Certificate request message or MIS_Push_Certificate indication message to the destination MN(s) or PoS(s), based on the ResponseFlag parameter.

7.4.23.2 MIS_Push_Certificate.indication

7.4.23.2.1 Function

This primitive is generated by an MISF to notify a local MIS user that an MIS_Push_Certificate request message has been received.

7.4.23.2.2 Semantics of service primitive

MIS_Push_Certificate.indication (
SourceIdentifier,
ResponseFlag,
Certificate
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Identifies the sender of the credential.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.
Certificate	CERTIFICATE	A PoS's X.509 certificate.

[a] If the ResponseFlag parameter is not present, the MISF shall generate a response message, otherwise the MISF generates either a response message or no response message, based on the ResponseFlag parameter.

7.4.23.2.3 When generated

This primitive is generated by an MISF when an MIS_Push_Certificate request message is received.

7.4.23.2.4 Effect on receipt

Upon receipt of this primitive, an MIS user on an MN or a PoS verifies a X.509 certificate in the Certificate. The serial number of the X.509 certificate and the result of verification are provided back to push requester via CertificateSerialNumber parameter and CertificateStatus parameter in MIS_Push_Certificate.response primitive. The CertificateStatus parameter value is set as follows. If the X.509 certificate is revoked by a certificate revocation list, then CertificateStatus is set to "Certificate Revoked," else if the X.509 is expired then CertificateStatus is "Certificate Expired," else if a signature in X.509 is not valid, then CertificateStatus is set to "Verification Failed," else CertificateStatus is set to "Certificate Valid." If the ResponseFlag parameter is not present, or ResponseFlag parameter is present and its value is '1', the MIS user shall generate an MIS_Push_Certificate.response primitive.

7.4.23.3 MIS_Push_Certificate.response

7.4.23.3.1 Function

This primitive is generated by an MIS user to acknowledge receipt of a credential from a PoS.

7.4.23.3.2 Semantics of service primitive

MIS_Push_Certificate.response (
DestinationIdentifier,
CertificateSerialNumber,
CertificateStatus
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies the sender of the credential.
CertificateSerialNumber	CERT_SERIAL_NUMBER	X.509 certificate subfield–serial number.
CertificateStatus	CERT_STATUS	Indicates whether a credential has been verified and is now in use by the recipient. One of the following values is used: 1: Certificate Valid 2: Certificate Revoked 3: Certificate Expired 4: Verification Failed

7.4.23.3.3 When generated

An MIS user generates this primitive after receiving and processing acredential.

7.4.23.3.4 Effect on receipt

An MIS_Push_Certificate response message is sent back to the sender of the credential to indicate a serial number in the credential and a status of the credential by providing the CertificateStatus. When CertificateStatus is “Certificate Valid,” the validated credential public keys should be utilized for multicast message exchange within their expiration period.

7.4.23.4 MIS_Push_Certificate.confirm

7.4.23.4.1 Function

This primitive is generated by an MISF that receives an MIS_Push_Certificate response message to indicate the status of the credential inspection.

7.4.23.4.2 Semantics of service primitive

MIS_Push_Certificate.confirm (
SourceIdentifier,
CertificateSerialNumber,
CertificateStatus
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Identifies the remote MISF that invoked MIS_Push_Certificate.response.
CertificateSerialNumber	CERT_SERIAL_NUMBER	X.509 certificate subfield–serial number.
CertificateStatus	CERT_STATUS	Indicates whether a credential has been verified and is now in use by the recipient.

7.4.23.4.3 When generated

The MISF that receives an MIS_Push_Certificate response message generates this primitive to indicate the serial number in the credential and status of the credential inspection.

7.4.23.4.4 Effect on receipt

If CertificateStatus is "Certificate Valid," then it indicates to the MIS user that a receiver of the MIS_Push_Certificate request message is capable of receiving signed multicast messages.

7.4.24 MIS_Revoke_Certificate

7.4.24.1 MIS_Revoke_Certificate.request

7.4.24.1.1 Function

This primitive is generated by a PoS used to revoke a credential.

7.4.24.1.2 Semantics of service primitive

MIS_Revoke_Certificate.request (
DestinationIdentifier,
ResponseFlag,
CertificateSerialNumberList,
CertificateRevocation,
IssuerName
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies an MISF or a group of MISF peers to revoke the credential.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.
CertificateSerialNumberList	CERT_SERIAL_NUMBER_INFO	List of revoked X.509 certificate subfield–serial number, Bloom Filter of revoked X.509 certificate subfield–serial number, or X.509 Certificate Revocation List.
CertificateRevocation	SIGNATURE	(Optional) Digital signature for a revoked X.509 certificate serial numbers generated by CA. This parameter shall be contained if and only if List or Bloom Filter of revoked X.509 certificate subfield–serial numbers is contained in CertificateSerialNumberList.
IssuerName	OCTET_STRING	(Optional) Distinguished name of the issuer of the revoked certificates.

[a] If the ResponseFlag parameter is not present, the MISF shall generate a request message, otherwise the MISF generates either a request or an indication message, based on the ResponseFlag parameter.

7.4.24.1.3 When generated

The MIS user generates this primitive to revoke credentials.

7.4.24.1.4 Effect on receipt

Upon receipt of this primitive, the MISF on the PoS generates and sends the corresponding MIS_Revoke_Certificate request message or MIS_Revoke_Certificate indication message to the destination MISF(s), based on the ResponseFlag parameter.

7.4.24.2 MIS_Revoke_Certificate.indication

7.4.24.2.1 Function

This primitive is generated by an MISF to revoke a credential stored in MN(s) and PoS(s).

7.4.24.2.2 Semantics of service primitive

MIS_Revoke_Certificate.indication (
SourceIdentifier,
ResponseFlag,
CertificateSerialNumberList,
CertificateRevocation,
IssuerName
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Specifies the remote MISF that invoked MIS_Revoke_Certificate.request primitive.
ResponseFlag[a]	RESPONSE_FLAG	(Optional) Flag that represents whether or not a response is needed.
CertificateSerialNumberList	CERT_SERIAL_NUMBER_INFO	List of revoked X.509 certificate subfield–serial number, Bloom Filter of revoked X.509 certificate subfield–serial number, or X.509 Certificate Revocation List.
CertificateRevocation	SIGNATURE	(Optional) Digital signature for a revoked X.509 certificate serial numbers generated by CA. This parameter shall be contained if and only if List or Bloom Filter of X.509 certificate subfield–serial numbers is contained in CertificateSerialNumberList.
IssuerName	OCTET_STRING	(Optional) Distinguished name of the issuer of the revoked certificates.

[a] If the ResponseFlag parameter is not present, the MISF shall generate a response message, otherwise the MISF generates either a response message or no response message, based on the ResponseFlag parameter.

7.4.24.2.3 When generated

This primitive is generated by an MISF on an MN or a PoS when receiving an MIS_Revoke_Certificate request message from a remote MISF.

7.4.24.2.4 Effect on receipt

Upon receipt of this primitive, if a CertificateRevocation is presented, an MIS user on an MN or a PoS verifies a signature in the CertificateRevocation, else the MIS user verifies a signature contained in X.509 Certificate Revocation List in the CertificateSerialNumberList. If the signature is valid and certificate serial numbers indicated by the CertificateSerialNumberList are present, then it deprecates the certificate specified by the CertificateSerialNumber and invokes an MIS_Revoke_Certificate.response primitive with CERT_STATUS “Certificate Revoked,” else if the certificate serial numbers indicated by the CertificateSerialNumberList are not present, it invokes an MIS_Revoke_Certificate.response primitive with CERT_STATUS “Not Present.” If the signature is not valid, it invokes an MIS_Revoke_Certificate.response primitive with CERT_STATUS “Verification Failed.” If the ResponseFlag parameter is not present, or ResponseFlag parameter is present and its value is ‘1’, the MIS user shall generate an MIS_Push_Group_Manipulate.response primitive.

7.4.24.3 MIS_Revoke_Certificate.response

7.4.24.3.1 Function

This primitive is generated by an MIS user to acknowledge receipt of a credential revocation request from a PoS.

7.4.24.3.2 Semantics of service primitive

MIS_Revoke_Certificate.response (
DestinationIdentifier,
CertificateStatus
)

Parameters:

Name	Data type	Description
DestinationIdentifier	MISF_ID	Specifies the remote MISF that invoked MIS_Revoke_Certificate.request primitive.
CertificateStatus	CERT_STATUS	Indicates whether a credential has been verified and is now in use by the recipient. One of the following values is used: 0: Not Present 2: Certificate Revoked 4: Verification Failed

7.4.24.3.3 When generated

This primitive is generated by an MIS user on an MN or a PoS when receiving an MIS_Revoke_Certificate.indication primitive.

7.4.24.3.4 Effect on receipt

Upon receipt of this primitive, an MIS user on an MN or a PoS deprecate the credential specified by the CertificateSerialNumber and invokes an MIS_Revoke_Certificate.confirm primitive.

7.4.24.4 MIS_Revoke_Certificate.confirm

7.4.24.4.1 Function

This primitive is generated by an MISF that receives an MIS_Revoke_Certificate response to indicate the status of the credential revocation.

7.4.24.4.2 Semantics of service primitive

MIS_Revoke_Certificate.confirm (
SourceIdentifier,
CertificateStatus
)

Parameters:

Name	Data type	Description
SourceIdentifier	MISF_ID	Identifies the remote MISF that invoked MIS_Revoke_Certificate.response.
CertificateStatus	CERT_STATUS	Indicates whether a credential has been revoked.

7.4.24.4.3 When generated

The MISF that receives an MIS_Revoke_Certificate response message generates this primitive to indicate the status of the credential revocation.

7.4.24.4.4 Effect on receipt

If CertificateStatus indicates success for all the MISF peers to which credential revocation request was sent, the PoS changes status of the credential to revoked.

7.5 MIS_NET_SAP primitive

7.5.1 MIS_TP_Data

7.5.1.1 General

The primitives associated with data transfers are as follows:

— MIS_TP_Data.request
— MIS_TP_Data.indication
— MIS_TP_Data.confirm

The MISF uses the MIS_TP_Data.request primitive to request that an MIS PDU be transported. The transport service provider uses the MIS_TP_Data.indication primitive to indicate the arrival of an MIS PDU. MIS_TP_Data.confirm primitive is used to acknowledge the successful transfer of the MIS PDU.

7.5.1.2 MIS_TP_Data.request

7.5.1.2.1 Function

This primitive is the request for transfer of an MIS PDU.

7.5.1.2.2 Semantics

```
MIS_TP_Data.request     (
                        TransportType,
                        SourceAddress,
                        DestinationAddress,
                        ReliableDeliveryFlag,
                        MISProtocolPDU
                        )
```

Parameters:

Name	Data type	Description
TransportType	TRANSPORT_TYPE	Identifies the protocol layer specific transport option.
SourceAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Source MISF.
DestinationAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Destination MISF.
ReliableDeliveryFlag	BOOLEAN	Indicate that the data is sent reliably and an error is generated if delivery fails. TRUE: Reliable delivery is required. FALSE: Reliable delivery is not required.
MISProtocolPDU	OCTET_STRING	MIS Protocol PDU to be transferred.

7.5.1.2.3 When generated

This primitive is used to request that an MIS PDU be transported to a remote MISF.

7.5.1.2.4 Effect on receipt

The receipt of this primitive causes the selected transport service provider to attempt to transport the MIS PDU.

7.5.1.3 MIS_TP_Data.indication

7.5.1.3.1 Function

This primitive is the indication of a received MIS PDU.

7.5.1.3.2 Semantics

MIS_TP_Data.indication (
TransportType,
SourceAddress,
DestinationAddress,
ReliableDeliveryFlag,
MISProtocolPDU
)

Parameters:

Name	Data type	Description
TransportType	TRANSPORT_TYPE	Identifies the protocol layer specific transport option.
SourceAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Source MISF.
DestinationAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Destination MISF.
ReliableDeliveryFlag	BOOLEAN	Indicate that the data is sent reliably and an error generated if delivery fails. TRUE: Reliable delivery is required. FALSE: Reliable delivery is not required.
MISProtocolPDU	OCTET_STRING	MIS protocol PDU received.

7.5.1.3.3 When generated

This primitive is used by the transport service provider to indicate that an MIS PDU has been received from a remote MISF.

7.5.1.3.4 Effect on receipt

The receipt of this primitive causes the MISF to receive the MIS PDU that was transported.

7.5.1.4 MIS_TP_Data.confirm

7.5.1.4.1 Function

This primitive is used to confirm an acknowledged transfer.

7.5.1.4.2 Semantics

MIS_TP_Data.confirm (
Status,
TransportType,
SourceAddress,
DestinationAddress
)

Parameters:

Name	Data type	Description
Status	STATUS	Status of operation.
TransportType	TRANSPORT_TYPE	Identifies the protocol layer specific transport option.
SourceAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Source MISF.
DestinationAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Destination MISF.

7.5.1.4.3 When generated

This primitive is passed from the transport service provider to the MISF to confirm that a request to transfer an MIS PDU succeeded.

7.5.1.4.4 Effect on receipt

Upon receipt of this primitive, the receiving MISF stops its retransmission timer for the corresponding request. When the MISF does not receive this primitive for a pre-defined time after transmitting an MIS_TP_Data.request with ReliableDeliveryFlag set to TRUE, the MISF attempts to retransmit the MIS_TP_Data.request.

8. Media independent service protocol

8.1 Introduction

The MIS function entities in MN and network entities communicate with each other using the MIS protocol messages specified in this clause. The MIS protocol defines message formats for exchanging these messages between peer MIS function entities. These messages are based on the primitives that are part of the MIS services.

8.2 MIS protocol description

8.2.1 MIS protocol transaction

The media independent service protocol defines a message exchange between two MISF entities to support remote MISF services. An MIS transaction is identified by a sequence of messages with the same Transaction-ID submitted to, or received from, one specific remote MISF ID.

At any given moment, an MIS node shall have no more than one transaction pending for each direction with a certain MIS peer. In other words, the MIS node shall wait until any pending outgoing transaction is completed before it creates another outgoing transaction for the same peer. Similarly, the MIS node shall wait until any pending incoming transaction is completed before it creates another incoming transaction for the same peer.

8.2.2 MIS protocol acknowledgement service

The acknowledgement service shall be used when the MIS transport used for remote communication does not provide reliable services. When the MIS transport is reliable, the use of the acknowledgement service is not needed. The acknowledgement service is particularly useful when the underlying transport used for remote communication does not provide reliable services. When the MIS transport is reliable, the acknowledgement service is optional. In case the destination of the communication is an MISF Group ID, the acknowledgement service shall not be used, even in cases when underlying transport is not reliable.

The source MISF requests for an acknowledgement message to ensure successful receipt of an MIS protocol message. This MIS message is used to acknowledge the successful receipt of an MIS protocol message at the destination MISF.

The MIS acknowledgement service is supported by the use of two bits of information that are defined exclusively for acknowledgement (ACK) usage in the MIS header. The ACK-Req bit is set by the source MIS node and the ACK-Rsp bit is set by the destination MIS node to utilize the acknowledgement service. It is expected that the underlying transport layer would take care of ensuring the integrity of the MIS protocol message during delivery.

When seeking acknowledgement service, the source MIS node shall start a retransmission timer after sending an MIS protocol message with the ACK-Req bit set and saves a copy of the MIS protocol message while the timer is active. The algorithm defined in IETF RFC 2988 is used to calculate the value of the retransmission timer. If the acknowledgement message is not received before the expiration of the timer, the source MIS node immediately retransmits the saved message with the same Message-ID and with the same Transaction-ID (with ACK-Req bit set). If the source MIS node receives the acknowledgement before the expiration of the timer on the first or any subsequent retransmitted attempt, then the source MIS node has ensured the receipt of the MIS packet and, therefore, resets the timer and releases the saved copy of the MIS protocol message. During retransmission, if the source MIS node receives the acknowledgement for any of the previous transmission attempts, then the source MIS node determines successful delivery of the message and does not have to wait for any further acknowledgements for the current message. The source MIS node retransmits an MIS protocol message with ACK-Req bit set until it receives an acknowledgment or the number of retransmissions reaches its maximum value. The maximum number of retransmissions is configurable through a parameter defined in the MIB (see Annex I). The source MIS node does not attempt to retransmit a message with same Message-ID and Transaction-ID when the ACK-Req bit was not set in the first MIS message. Implementations should consider adjusting the retransmission time-out (RTO) when operating over links with power save mobile nodes.

When a destination MIS node receives an MIS protocol message with the ACK-Req bit set, then the destination MIS node returns an MIS message with the ACK-Rsp bit set and copying the Message-ID and Transaction-ID from the received MIS protocol message. The MIS message with the ACK-Rsp bit set has only the MIS header and no other payload. In instances where the destination MIS node immediately processes the received MIS protocol message and a response is immediately available, then the ACK-Rsp bit is set in the corresponding MIS protocol response message.

The destination MIS node responds with an acknowledgement message for duplicate MIS messages (messages with same message-id and transaction-ID) that have the ACK-Req bit set. However, the destination MIS node does not process these duplicate messages if it has already done so. If a destination MIS node receives an MIS protocol message with no ACK-Req bit set, then no action is taken with respect to the acknowledgement service.

In all other cases, the MIS protocol message in a transaction is processed only once at the destination MIS node, irrespective of the number of received messages with the ACK-Req bit set. The destination MIS node sets the ACK-Rsp bit in an MIS protocol response message and additionally requests acknowledgement by setting the ACK-Req bit for the same MIS protocol response message.

In case an MIS protocol message with destination MISF Group ID is received with the ACK-Req bit set, the received station shall ignore this bit.

8.2.3 MIS protocol transaction state diagram

8.2.3.1 State machines

A node that has a new available message to send related to a new transaction is called a transaction source and starts the transaction source state machine. In the same manner, a node that receives a message related to a new transaction is called a transaction destination node and starts the destination transaction state machine.

If the ACK feature is being used by the source and/or destination transaction node, the ACK-requestor and/or ACK-responder state machine is started (specific conditions specified below). The ACK related state machine is run in parallel to the transaction source/destination state machines.

Each transaction is represented in an MISF by an instance of the transaction source or destination state machine. Optionally, each transaction also has one instance of ACK-requestor or one instance of ACK-responder state machine, or both.

All instances of the state machines related to one transaction have access to inter-state-machine variables, constants, and procedures, which are not accessible by the state machines related to other transactions. The inter-state-machine variables allow communications between state machines for a given transaction. There are no cases where two or more state machines for a given transaction write the same inter-state-machine variable at the same time. Intra-state-machine variables, constants, and procedures are accessible within a single state machine for a given transaction.

Figure 21 illustrates the interaction of transaction source/destination state machines with the ACK-related state machines.

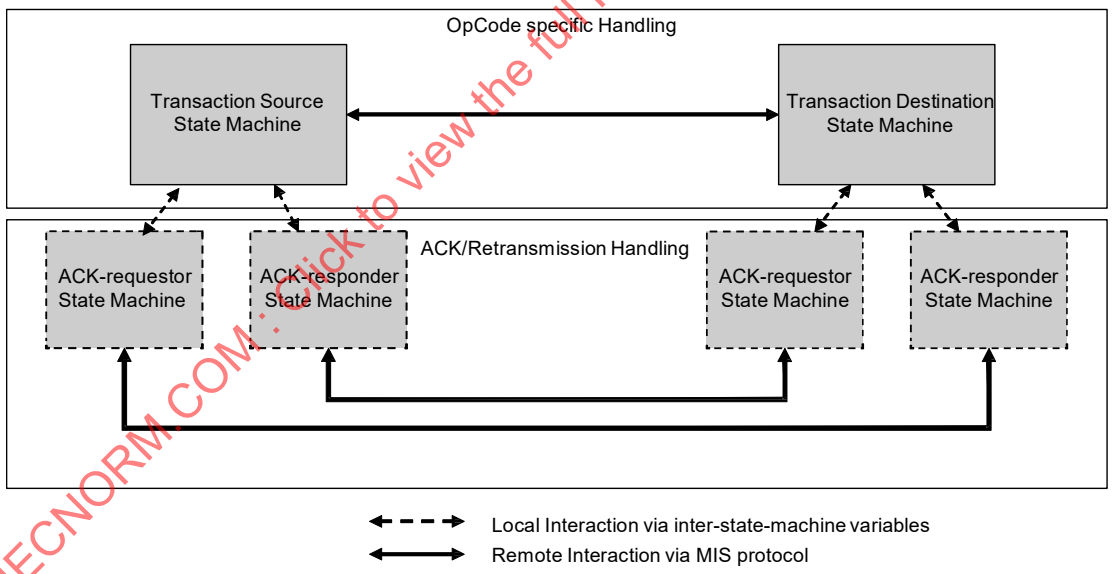


Figure 21—State machines interactions

8.2.3.2 Notational conventions used in state diagrams

State diagrams are used to represent the operation of an MIS transaction as a group of connected, mutually exclusive states. At any given time, only one state of each state machine is active per transaction instance.

Each state is represented in the state diagram as a rectangular box, divided into two parts by a horizontal line. The upper part contains the state identifier, written in uppercase letters. The lower part contains any procedures that are executed on entry to the state.

All permissible transitions between states are represented by arrows, the arrowhead denoting the direction of the possible transition. Labels attached to arrows denote the condition(s) that shall be met in order for the transition to take place.

A transition that is global in nature (i.e., a transition that occurs from any of the possible states if the condition attached to the arrow is met) is denoted by an open arrow; i.e., no specific state is identified as the origin of the transition.

On entry to a state, the procedures defined for the state (if any) are executed exactly once, in the order that they appear on the page. Each action is deemed to be atomic; i.e., execution of a procedure completes before the next sequential procedure starts to execute. No procedures execute outside of a state block. On completion of all of the procedures within a state, all exit conditions for the state (including all conditions associated with global transitions) are evaluated continuously until such a time as one of the conditions is met. All exit conditions are regarded as Boolean expressions that evaluate to TRUE or FALSE; if a condition evaluates to TRUE, then the condition is met.

The label UCT denotes an unconditional transition (i.e., UCT always evaluates to TRUE).

A variable that is set to a particular value in a state block retains this value until a subsequent state block executes a procedure that modifies that value.

Should a conflict exist between the interpretation of a state diagram and either the corresponding transition tables or the textual description associated with the state machine, the state diagram takes precedence.

The interpretation of the special symbols and operators used in the state diagrams is defined in Table 17; these symbols and operators are derived from the notation of the “C” programming language, ANSI X3.159.

Table 17—State machine symbols

Symbol	Interpretation
( )	Used to force the precedence of operators in Boolean expressions and to delimit the argument(s) of actions within state boxes.
;	Used as a terminating delimiter for actions within state boxes. Where a state box contains multiple actions, the order of execution follows the normal English language conventions for reading text.
=	Assignment action. The value of the expression to the right of the operator is assigned to the variable to the left of the operator. Where this operator is used to define multiple assignments, (e.g., a = b = X) the action causes the value of the expression following the right-most assignment operator to be assigned to all of the variables that appear to the left of the right-most assignment operator.
!	Logical NOT operator.
&&	Logical AND operator.
	Logical OR operator.
(statement1 ? statement2: statement3)	Conditional action. If statement1 evaluates to TRUE, then statement2 is executed. Otherwise statement3 is executed.
==	Equality. Evaluates to TRUE if the expression to the left of the operator is equal in value to the expression to the right.
<	Less than. Evaluates to TRUE if the value of the expression to the left of the operator is less than the value of the expression to the right.
++	Arithmetic increment by one operator.

8.2.3.3 Inter-state-machine variables

Inter-state-machine variables are available for use by more than one state machine related to one transaction instance and are used to perform inter-state-machine communication and initialization functions within that transaction.

Exported variables are inter-state-machine variables that are also readable and writable from entities external to the state machines. The inter-state-machine and exported state machine variables are specified in Table 18 and Table 19, respectively.

Table 18—Inter-state-machine variables

Name	Type	Description
Opcode	OPCODE	An Opcode.
MID	MID	A message identifier.
AckRequestorStatus	ENUMERATED	Indicates the status of the ACK requestor state machine. This variable is initialized by the transaction source state machine or transaction destination state machine and changed by the ACK requestor state machine. The following values are valid: 1: ONGOING 2: SUCCESS 3: FAILURE
TransactionStopWhen	UNSIGNED_INT(1)	A timer to stop the transaction.
RetransmissionWhen	UNSIGNED_INT(1)	A timer to retransmit a message.

Table 19—Exported state machine variables

Name	Type	Description
TID	TID	A transaction identifier.
MyMisfID	MISF_ID	The MISF ID of this MIS node.
PeerMisfID	MISF_ID	The MISF ID of the peer MIS node.
MsgIn	MIS_MESSAGE	A valid incoming message received from a remote MISF. An incoming message is valid in terms of state machine operation if the message has the Operation Code of the value Request (0x1), Response (0x2), or Indication (0x3).
MsgInAvail	BOOLEAN	This variable is set to TRUE by an entity external to the state machines when a valid incoming message is available for a transaction. The transaction corresponds to an instance of either Transaction Source State Machine or Transaction Destination State Machine depending on the Operation Code, destination identifier TLV, and ACK-Rsp bit of the message as shown in Table 20. The correspondence between an incoming message and a transaction is based on TID, MyMisfID, and PeerMisfID variables of Transaction Source or Destination State Machine against the Transaction ID field, destination identifier TLV, and source identifier TLV of the incoming message, respectively. This variable is initialized to FALSE by the external entity. This variable is set to FALSE by the state machines once the incoming message has been processed. It is the responsibility of the external entity to set this variable to TRUE such that this MIS node has no more than one transaction pending for each direction with a certain MIS peer.

Table 19—Exported state machine variables *(continued)*

Name	Type	Description
MsgOut	MIS_MES SAGE	A valid outgoing message generated by the local MISF to be sent to the remote MISF. An outgoing message is valid in terms of state machine operation if the message has the Operation Code of the value Request (0x1), Response (0x2), or Indication (0x3).
MsgOutAvail	BOOLEAN	This variable is set to TRUE by an entity external to the state machines or by Transaction Source or Destination State Machine when a valid outgoing message is available for a transaction. The transaction corresponds to an instance of either Transaction Source State Machine or Transaction Destination State Machine depending on the Operation Code and destination identifier TLV of the message as shown in Table 21. The correspondence between an outgoing message and a transaction is made based on matching TID, MyMisfID, and PeerMisfID variables of Transaction Source or Destination State Machine instances against the Transaction ID field, source identifier TLV, and destination identifier TLV of the outgoing message, respectively. This variable is initialized to FALSE by the external entity. It is the responsibility of the external entity to set this variable to TRUE such that this MIS node has no more than one transaction pending for each direction with a certain MIS peer.
TransactionStatus	ENUMERATED	Indicates the status of the transaction. This variable is written by the state machine and read by the MISF. The following values are valid: 1: ONGOING 2: SUCCESS 3: FAILURE
StartAckRequestor	BOOLEAN	This variable is initialized to FALSE by an external entity. The instance of ACK-requestor state machine is started when this variable is set to TRUE by its associated transaction source or destination state machine.
StartAckResponder	BOOLEAN	This variable is initialized to FALSE by an external entity. The instance of ACK-responder state machine is started when this variable is set to TRUE by its associated transaction source or destination state machine.

Table 20—State Machines to be searched for incoming message

Operation code	ACK-Rsp bit	Contains MIS service specific TLVs or a fragment payload	State machine instances to be searched: transaction source state machine (S) or transaction destination state machine (D)
Request (0x1)/Indication (0x3)	0	—	D
	1	—	S
Response (0x2)	0	—	S
	1	Yes	S
		No	D

Table 21—State Machines to be searched for outgoing message

Operation code	State machine instances to be searched: transaction source state machine (S) or transaction destination state machine (D)
Request (0x1)/Indication (0x3)	S
Response (0x2)	D

8.2.3.4 Inter-state-machine procedures

a) **BOOLEAN Process(MIS_MESSAGE)**—This procedure processes the incoming message passed as an input variable. A value of TRUE is returned if an outgoing message is available in response to the incoming message. Otherwise, a value of FALSE is returned.

b) **void Transmit(MIS_MESSAGE)**—This procedure transmits the message passed as the input variable.

c) **BOOLEAN IsMulticastMsg(MIS_MESSAGE)**—This procedure outputs TRUE if the input message has an MISF Group ID in the destination MISF_ID. Otherwise, it outputs FALSE.

d) **MISF_ID SrcMISF_ID(MIS_MESSAGE)**—This procedure obtains a source identifier TLV from the message passed as the input and returns the value of the TLV.

e) **MISF_ID DstMISF_ID(MIS_MESSAGE)**—This procedure obtains a destination identifier TLV from the message passed as the input and returns the value of the TLV.

f) **void SetMISF_ID(MIS_MESSAGE, MISF_ID, MISF_ID)**—This procedure inserts a source identifier TLV and a destination identifier TLV into the MIS message. The first MISF_ID is used as the value of the source identifier TLV. The second MISF_ID is used as the value of the destination identifier TLV.

8.2.3.5 Inter-state-machine constants

a) **TransactionLifetime**—The maximum time from the initiation of a transaction until its termination.

b) **Request**—An OPCODE value of 0x1.

c) **Response**—An OPCODE value of 0x2.

d) **Indication**—An OPCODE value of 0x3.

8.2.3.6 Timers

The timers defined for these state machines are decremented, if their value is non-zero, by the operation of Transaction Timers state machine. All timers have a resolution of one second, i.e., the initial values used to start the timers are integer values, and they represent the timer period as an integral number of seconds.

8.2.3.6.1 Intra-state machine variables and constants

a) **Tick**—This variable is set in response to a regular one-second tick generated by an external system clock function. Whenever the system clock generates a one-second tick, the tick variable is set to TRUE. The variable is set to FALSE by the operation of the state machine. The operation of the system clock functions is not otherwise specified by the standard.

b) **void dec(Timer)**—This procedure decrements the timer only if its value is greater than 0.

8.2.3.6.2 Transaction timers state machine

The transaction timers state machine (see Figure 22) for a given transaction is responsible for decrementing the timer variables for this transaction each second, in response to an external system clock function. The timer variables are used, and set to their initial values, by the operation of the individual state machines for the transaction.

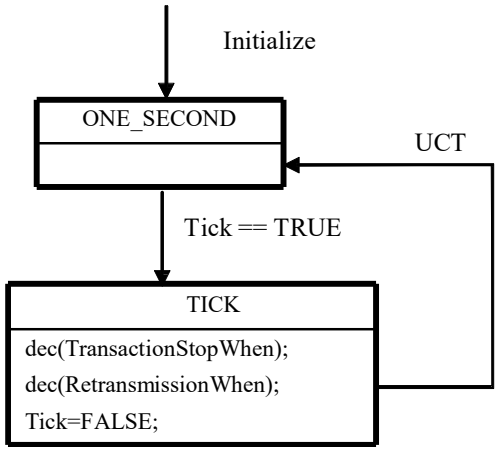


Figure 22—Transaction timers state machine

8.2.3.7 Transaction source and destination state machines

8.2.3.7.1 Intra-state-machine variables

a) **IsMulticast**—This variable's type is BOOLEAN. When its value is TRUE, it indicates that a message has a destination MISF Group ID. Otherwise, its value is FALSE.

b) **ResponseSent**—This variable's type is BOOLEAN. When its value is TRUE, it indicates that a Response message has been sent. Otherwise, its value is FALSE.

8.2.3.7.2 Intra-state-machine procedures

a) **ResponseReceived**—This variable's type is BOOLEAN. When its value is TRUE it indicates that a Response message has been received. Otherwise, its value is FALSE.

b) **TID NewTID(void)**—This procedure generates a new transaction ID for the transaction generated by the new available message.

8.2.3.7.3 Transaction source state machine

The transaction source state machine (see Figure 23) is started, and related transaction initiated, when a message related to a new transaction is available to be sent (MsgOutAvail is TRUE). The transaction terminates when it transits to the SUCCESS state and any ACK-related state machines if started were terminated; or if it transits to the FAILURE state. An instance of transaction source state machine ceases to exist once the value of TransactionStatus is set to either SUCCESS OR FAILURE.

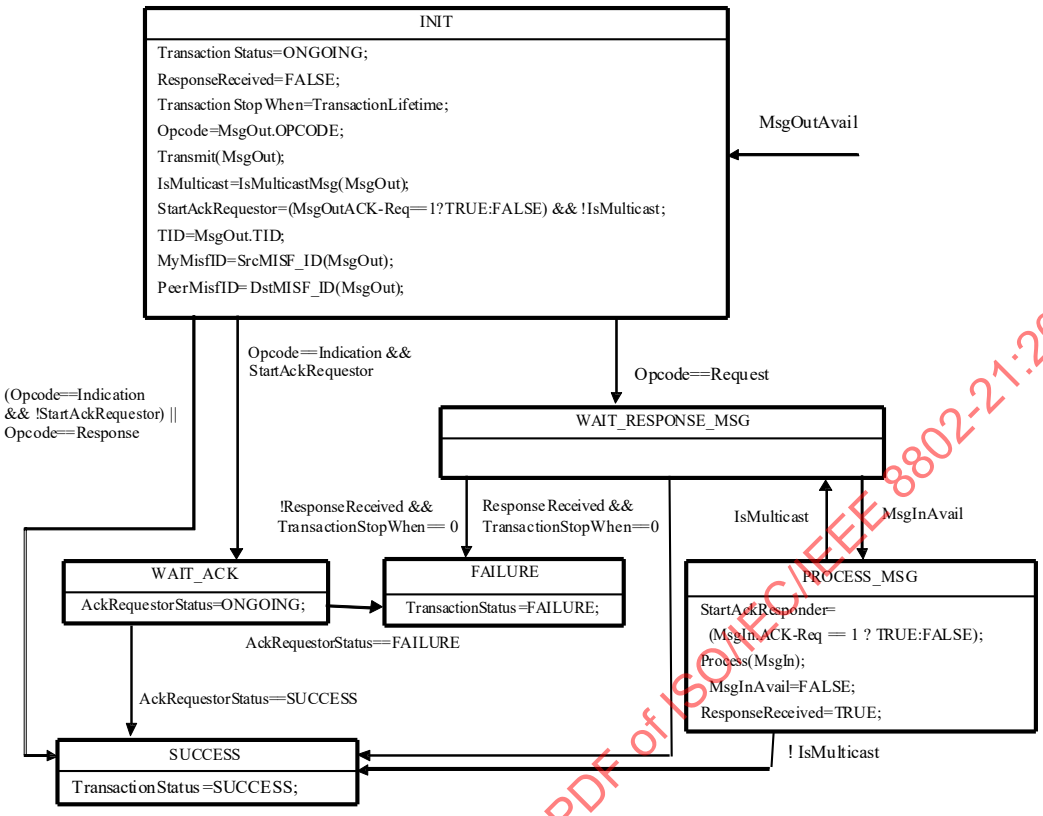


Figure 23—Transaction source state machine

8.2.3.7.4 Transaction destination state machine

The transaction destination state machine (see Figure 24) is started, and related transaction initiated, when a message related to a new transaction is received (MsgInAvail is TRUE).

The transaction terminates when it transits to the FAILURE state or SUCCESS state and any ACK-related state machines, if started, were terminated. An instance of transaction destination state machine ceases to exist once the value of TransactionStatus is set to either SUCCESS or FAILURE.

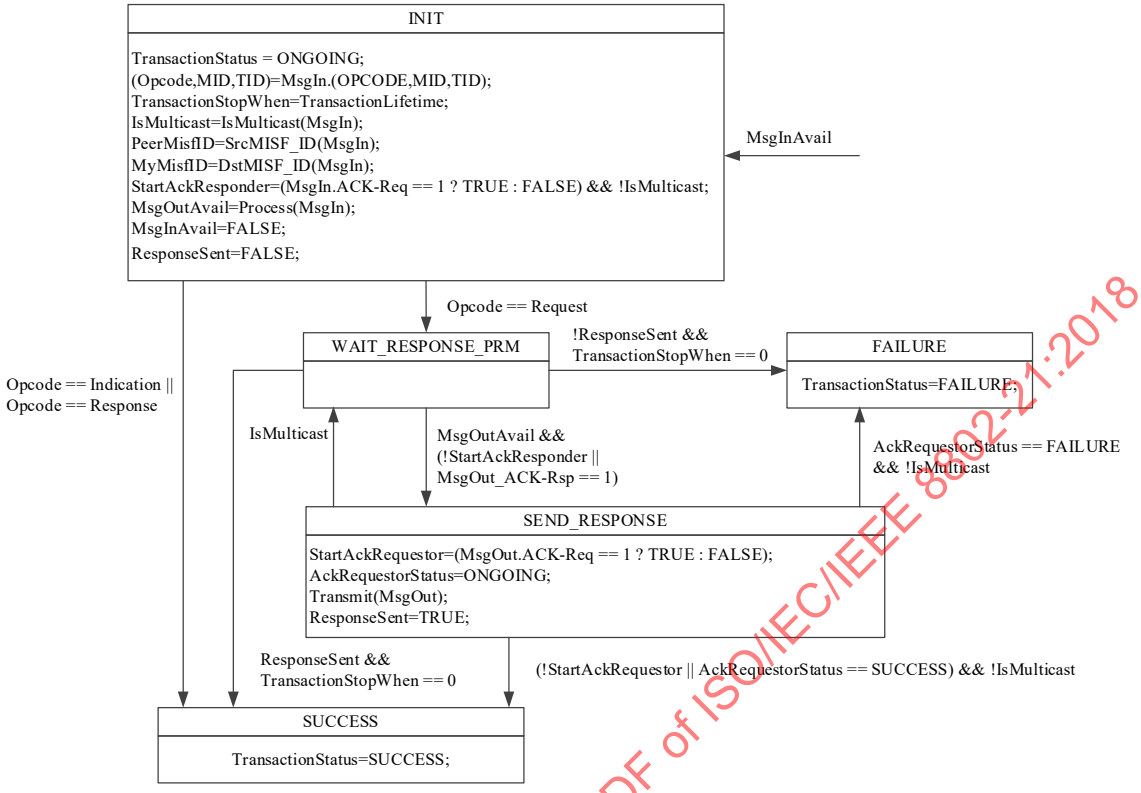


Figure 24—Transaction destination state machine

8.2.3.8 ACK-related state machines

The ACK-requestor state machine is started when the StartAckRequest variable turns TRUE and ACK-responder state machine is started when StartAckResponder variable turns TRUE.

8.2.3.8.1 Intra-state machine variables

a) **DUP**—This variable is of type MIS_MESSAGE and represents an MIS message that has already been sent. This variable is used within ACK Responder state machine.

b) **ACK**—This variable is of type MIS_MESSAGE and represents an MIS message with the ACK-Rsp bit set and the same message ID and transaction ID as the MIS message it acknowledges. This variable is used within ACK Responder state machine.

c) **RtxCtr**—This variable is of type UNSIGNED_INT(1) and represents a number of retransmissions of a specific message. This variable is used within ACK Requestor state machine.

8.2.3.8.2 Intra-state-machine constants

a) **RetransmissionInterval**—The time interval between two subsequent transmissions of a specific message.

b) **MaxRtxCtr**—The maximum number of times that a message is retransmitted, if retransmission condition occurs.

The maximum number of retransmissions, and the retransmission interval depends on the characteristics of the underlying transport. These configuration parameters are defined in an MIB, see Annex I.

Note that the maximum number of retransmissions is bounded by the transaction lifetime.

8.2.3.8.3 ACK requestor state machine

The ACK requestor state machine (see Figure 25) is started when the StartAckRequestor variable turns to TRUE in a source or destination transaction state machine. This state machine uses the inter-state-machine variables set by the originating state machine. This state machine terminates when it transits to the FAILURE state or SUCCESS state. An instance of ACK requestor state machine ceases to exist once the AckRequestorStatus is set to either SUCCESS or FAILURE state or its associated transaction source or transaction destination state machine ceases to exist.

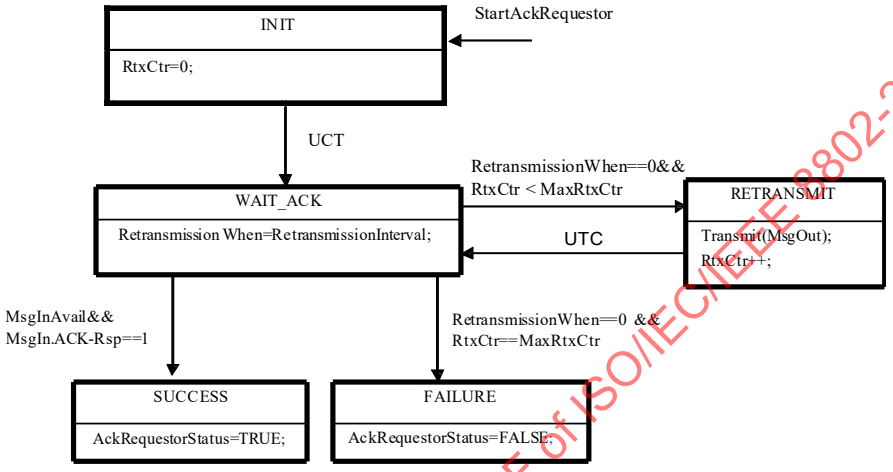


Figure 25—ACK requestor state machine

8.2.3.8.4 ACK responder state machine

The ACK responder state machine (see Figure 26) is started when the StartAckResponder variable turns to TRUE in a source or destination transaction state machine. This state machine uses the inter-state-machine variables set by the originating state machine. An instance of ACK responder state machine ceases to exist once its associated transaction source or transaction destination state machine ceases to exist.

8.2.4 Other considerations

8.2.4.1 Congestion control and load management

The MIS protocol does not provide direct support for congestion control. Therefore, it is recommended to run the MIS protocol over congestion-aware transport layers.

In order to help prevent congestion, flow control mechanisms are implemented at the MISF. A single rate limiter applies to all traffic (for all interfaces and message types). It applies to retransmissions, as well as new messages, although an implementation can choose to prioritize one over the other. When the rate limiter is in effect, MIS messages are queued until transmission is re-enabled, or an error condition is indicated back to local upper layer applications. The rate limiting mechanism is implementation specific, but it is recommended that a token bucket limiter, as described in IETF RFC 4443, be used.

When an MISF suffers from overload, it drops requests from MIS requestors. For example, messages could be dropped from a particular requestor if that requestor could be established as the origin of a denial of service attack. Any reliable delivery function indicates a flow control back to the requestor, and an MISF invokes flow control toward a specific requestor when overloaded with reliably delivered messages.

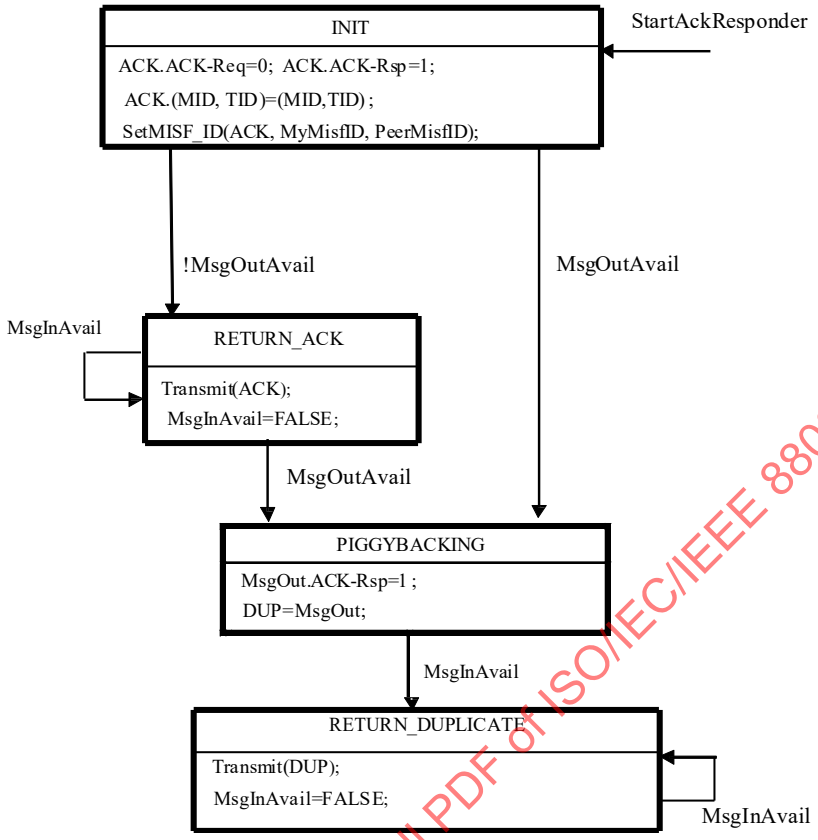


Figure 26—ACK responder state machine

8.2.4.2 Reliability

MIS protocol messages are delivered via media dependent transport. To help ensure proper operation, a reliable message delivery service is required. If the media dependent transport is unreliable, then the Acknowledgement Service shall be enabled, as specified in 8.2.2. If the media dependent transport is reliable, then the Acknowledgement Service is not mandatory.

A reliable media dependent transport is one that exhibits a message loss rate of less than 0.01%.

8.2.4.3 MISF discover

8.2.4.3.1 General

The MISF discovery refers to the procedure that allows one MISF to discover its peer MISFs (e.g., an MN discovers available peer MISFs in an access network). It is possible to perform MISF discovery either at layer 2 or layer 3. MISF discovery at L2 is performed either in media-specific manner (e.g., using IEEE 802.11 Beacon frames, IEEE 802.16 DCD) or using multicast data frames as described in 8.2.4.3.2 and 8.2.4.3.3. MISF discovery mechanisms at layer 3 are defined in the IETF RFC 5677, IETF RFC 5678, and IETF RFC 5679.

8.2.4.3.2 Combined MIS function discovery and capability discovery over data plane

Combined MIS function discovery and capability discovery is performed to discover the MISF ID, the peer MISF transport address, and MISF capabilities at the same time. As stated in 6.2.3, MISF Discovery can be implicitly performed using the MIS capability discovery when both MIS nodes are residing in the same

multicast domain (where an MIS node's multicast data frame is delivered using a group MAC address). If MISF ID and transport address are known (e.g., pre-configured) MISF uses MIS_Capability_Discover messages to discover MISF capabilities only. The following subclauses refer to the MIS capability discovery both as a means to discover the MISF and its capabilities.

8.2.4.3.3 Unsolicited MIS capability discovery

An MISF discovers peer MISF entities and their capabilities by listening to media-specific broadcast control messages. For example, by listening to a media-specific broadcast message such as a Beacon frame in IEEE Std 802.11-2012 or a DCD in IEEE Std 802.16-2012, link layers on an MN then forward the detected MIS capabilities to its MISF.

8.2.4.3.4 Solicited MIS capability discovery

An MISF (the requestor) discovers its peer MIS functions and capabilities sending an MIS_Capability_Discover request message to either its network multicast address with an MISF Group ID or to a unicast address with a known MISF ID. Network multicast address is used when the requester is either a mobile node (MN) or a network entity that does not have the destination MISF ID. Only MIS network entities respond to a multicast MIS_Capability_Discover request.

When a peer MIS function (the responder) receives the MIS_Capability_Discover request message, it sends MIS_Capability_Discover response message back to the requestor. The response is sent by using the same transport type over which the request message was received. When the requestor receives the unicast MIS_Capability_Discover response message, it learns the responder's MISF ID by checking the source ID of MIS_Capability_Discover response.

For complete operation, the requestor sets a timer at the time of sending an MIS_Capability_Discover request during which time the requestor is in waiting state for a response from the responder. When the response message is received while the timer is running, the requestor stops the timer and finishes the MIS function and capability discovery procedure. When the timer expires without receiving a response message, the requestor tries the combined MIS function discovery and capability discovery procedure by using a different transport or terminates the MIS function and capability discovery procedure.

If the MIS capability discovery is invoked upon receiving MIS capability advertisement in unauthenticated state through media-specific broadcast messages, such as beacon frames and DCD, destination MISF ID is filled with MISF Broadcast ID and this message is transmitted over the control plane using an L2 management frame, such as an IEEE 802.11 management action frame or an IEEE 802.16 MAC management message. This message contains the SupportedMISEventList, SupportedMISCommandList, SupportedISQueryTypeList, and SupportedTransportList TLVs to enable the receiving MISF to discover the sending MISF's capability. Therefore, peer MISF entities can discover each other's MIS capability by one MIS protocol message transaction. When the requestor receives the unicast MIS_Capability_Discover response message, which is embedded in the media-specific control message, it retrieves the responder's MISF ID by checking the source of the MIS_Capability_Discover response message.

8.3 MIS protocol identifier

The following identifiers are used in MIS protocol messages:

— MISF ID
— Transaction ID

8.3.1 MISF ID

MISF Identifier (MISF ID) is an identifier that is required to uniquely identify a specific MISF or a group of MISF peers for delivering the MIS services. MISF ID is used in all MIS protocol messages. This enables the MIS protocol to be transport agnostic.

MISF ID is assigned to the MISF during its configuration process. The configuration process is outside the scope of the standard.

MISF Broadcast ID is defined as an MISF Group ID of zero length. An MISF Broadcast ID (zero length) shall be used in an MIS message when destination MISF ID is not known to a source MISF. MISF Group ID is used when a message is addressed to a group of MISF peers.

In addition, the following rules apply to the case of messages addressed to an MISF Group ID:

- Multicast transmission is not allowed for MIES. No MIES primitive shall be generated with the destination MISF Group ID.
- Multicast transmission in general is not allowed for messages sent by the MN except for the MIS capability discover request message. Hence, commands in the form of MIS_MN_* (‘*’ means wildcard) cannot use multicast transmission.
- Multicast transmission is not allowed for MIS_NET_SAP primitives.
- Multicast transmission is not allowed for MIS_LINK_SAP primitives.

8.3.2 MISF group addressed message

The following MIS messages may use an MISF Group ID including MISF Broadcast ID as their destination identifier. The only allowed multicast transmission is when the message is sent by a PoS, although a message can be sent by a PoS and an MN:

a) MIS Messages for Management Service:
   - MIS_Registration request
   - MIS_DeRegister request
   - MIS_Capability_Discover
   - MIS_Push_Group_Manipulate request
   - MIS_Push_Group_Manipulate indication
   - MIS_Net_Push_Certificate request
   - MIS_Push_Certificate request
   - MIS_Revoke_Certificate request
   - MIS_Pull_Certificate request
   - MIS_Event_Subscribe request
   - MIS_Event_UnSubscribe request

b) MIS Messages for Command Service:
   - MIS_Link_Get_Parameters request
   - MIS_Link_Configure_Thresholds request/indication
   - MIS_Link_Actions request/indication
   - MIS_Configuration_Update request/indication

c) MIS Messages for Information Service:
   - MIS_Push_Information indication

The MISF ID is of type MISF_ID. (See E.3.11.)

8.3.3 Transaction ID

Transaction Identifier (Transaction ID) is an identifier that is used to match a request message with its corresponding response message. This identifier is also required to match each request, response, or indication message and its corresponding acknowledgment. This identifier is created at the node initiating the transaction and it is carried over within the fixed header part of the MIS protocol frame.

Transaction ID is defined as a 12 bit long unsigned integer whose value is unique among all the pending transactions between a given pair of the sender and receiver. For example, this could be an integer that starts from a random initial value and incremented by one (modulo 2^{12}) every time a new Transaction ID is generated.

8.4 MIS protocol frame format

8.4.1 General frame format

In MIS protocol messages, all TLV definitions are always aligned on an octet boundary and hence no padding is required. An MIS protocol payload carries a source MISF identifier TLV and a destination MISF identifier TLV followed by MIS service specific TLVs.

Figure 27 shows the components of the MIS protocol frame.

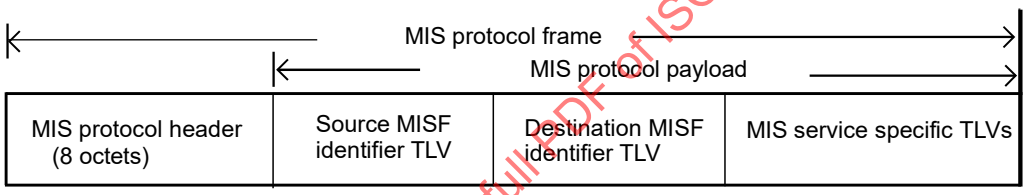


Figure 27—MIS protocol general frame format

The MIS protocol header (see Figure 28) carries the essential information that is present in every frame and is used for parsing and analyzing the MIS protocol frame.

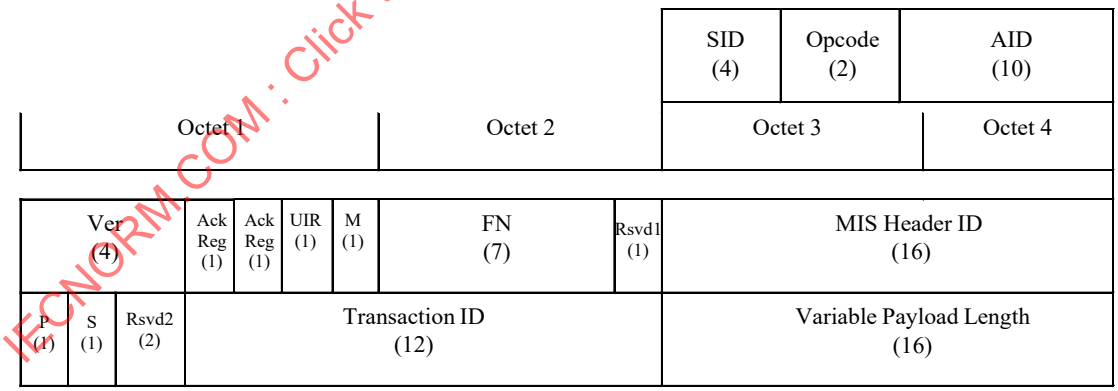


Figure 28—MIS protocol header format

Table 22 shows the description of the header fields.

Table 22—Description of MIS protocol header fields

Field name	Size (bits)	Description
Version	4	This field is used to specify the version of MIS protocol used. 0: Not to be used 1: First version 2 to 15: (Reserved) The version number shall be incremented only when a fundamental incompatibility exists between a new revision and the prior edition of the standard. An MIS node that receives an MIS message with a higher version number than it supports shall discard the frame without indication to the sending MIS node.
ACK-Req	1	This field is used for requesting an acknowledgement for the message.
ACK-Rsp	1	This field is used for responding to the request for an acknowledgement for the message.
Unauthenticated information request (UIR)	1	This field is used by the MIS Information Service to indicate if the protocol message is sent in pre-authentication/pre-association state. In this case, the length of the response message is limited. The UIR bit should be set to '1' by the originator when making an MIS information service request over a certain link in the unassociated/unauthenticated or unregistered state. In all other cases, this bit is set to '0'.
More fragment (M)	1	This field is used for indicating that the message is a fragment to be followed by another fragment. It is set to '0' for a message that is not fragmented and for the last fragment. The two 0 valued conditions are differentiated by the FN field. It is set to '1' for a fragment that is not the last one.
Fragment number (FN)	7	This field is used for representing the sequence number of a fragment. The fragment number starts from 0. The maximum fragment number is 127. This field is set to '0' for a message that is not fragmented.
Reserved1	1	This field is intentionally kept reserved. When not used, this bit is set to '0'.
MIS message ID (MID)	16	Combination of the following 3 fields.
—Service identifier (SID)	4	Identifies the different MIS services; possible values are as follows: 1: Service Management 2: Event Service 3: Command Service 4: Information Service
—Operation code (Opcode)	2	Type of operation to be performed with respect to the SID; possible values are as follows: 1: Request 2: Response 3: Indication
—Action identifier (AID)	10	This indicates the action to be taken with regard to the SID (see Table K.1 for AID assignments).
Proactive authentication (P)	1	This field is used for indicating that the message is a proactive authentication message.
Security association (S)	1	This field is used for indicating that a security association exists and the message is protected.
Reserved2	2	This field is intentionally kept reserved. When not used, all the bits of this field are to be set to '0'.
Transaction ID	12	This field is used for matching Request and Response, as well as matching Request, Response, and Indication to an ACK.
Variable payload length	16	Indicates the total length of the variable payload embedded in this MIS protocol frame. The length of the MIS protocol header is NOT included.

8.4.2 Protected MIS protocol frame format

In an MIS header the following two bits are used to indicate that an MIS PDU is protected and/or is used to carry a proactive authentication message.

a) P bit—Setting P bit to '1' indicates that the message carries a proactive authentication message.
b) S bit—Setting S bit to '1' indicates that an MIS security association exists and the service specific TLVs are protected.

A protected MIS PDU is an MIS PDU that has an MIS header with S bit set to '1' indicating that the MIS service specific TLVs in this PDU are encrypted and/or the PDU is digitally signed. When the MIS service specific TLVs in this PDU are encrypted, each security association is defined for a pair or group of MISFs and is identified by a security association identifier (SAID). In this case, an SAID TLV shall be carried in the PDU. Source and destination MISF identifier TLVs may not be present when an SA is defined for a pair of MISFs. When a PDU is digitally signed, a signature TLV shall be carried in the PDU. A signature TLV should be used for multicast MIS messages in order to provide source origin authentication for multicast MIS messages. Otherwise, a message alternation attack by an insider who has a GKB-generated MIS SA is possible even if the multicast MIS message is integrity protected by the group key corresponding to the GKB-generated MIS SA. On the other hand, a message alternation attack by an outsider who does not have a GKB-generated MIS SA is not possible if the multicast MIS message is integrity protected by the GKB-generated MIS SA but is not protected by a signature TLV. Especially, a message protected by a GKB-generated MIS SA for a two-member group provides the source origin authentication without a signature TLV. Figure 29 shows a protected MIS protocol frame. Table 23 shows valid combinations of S bit and security-related TLVs.

Table 23—Valid combination of S-bit and security-related TLVs

S bit	Source and destination MISF identifier TLVs	SAID TLV	Security TLV or service specific TLVs	Signature TLV
0	Present	Not present	Service specific TLVs	Not present
1	Present	Not present	Service specific TLVs	Present
1	May or may not be present	Present && ID_TYPE ≠ 2	Security TLV	Not present
1	Present	Present && ID_TYPE = 2	Security TLV	May not be present

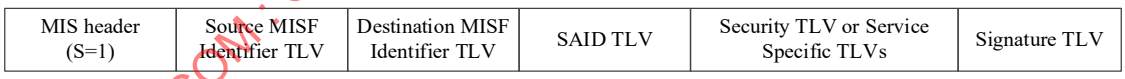


Figure 29—Protected MIS frame format

8.4.2.1 MIS PDU protected by (D)TLS

This clause specifies MIS message protection using transport layer security (TLS) (IETF RFC 5246) or datagram transport layer security (DTLS) (IETF RFC 6347).

The transport protocol for (D)TLS is the MIS protocol. When the MIS protocol transport is reliable, TLS is used. Otherwise, DTLS is used. The transport protocol entities to be associated with a TLS session are MISF peers and are identified by MISF identifiers. The TLS handshake takes place over the MIS protocol and as a result, an MIS SA that contains TLS master key and its child keys, TLS random values and the TLS ciphersuite negotiated in the TLS handshake is established between the peers. The detailed description about the protocol interface of using (D)TLS is provided in 9.1.

The structure of an MIS PDU during a TLS handshake is shown in Figure 30.

MIS header	Source MISF identifier TLV	Destination MISF identifier TLV	Security TLV (TLS record type = handshake/change cipher)

Figure 30—MIS PDU during TLS handshake

Once a (D)TLS handshake is completed, an MIS SA is established, which is determined by the ciphersuite negotiated in the (D)TLS handshake. The structure of protected MIS PDU, when an MIS SA exists, is shown in Figure 31, where the unprotected MIS service specific TLVs are carried and protected as (D)TLS application data. An MIS header has the S bit set to '1'. The TLS session ID assigned through TLS handshake is contained in the SAID TLV. The TLS provides integrity protection, encryption, or both. If it is integrity protected, then a message integrity code (MIC) is also included in the security TLV. In this standard, the message integrity code is the same as the message authentication code, for which the acronym MAC is already used for media access control.

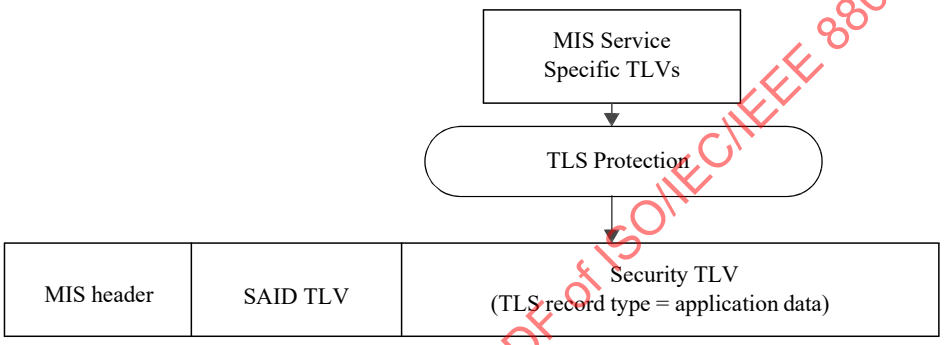


Figure 31—MIS PDU in existence of MIS SA by TLS

The MIS message protection procedure is specified in 9.3.

A signature TLV shall not be carried when MIS PDU is protected by (D)TLS (IETF RFC 6347). Note that an MIS PDU protected by (D)TLS should only be used for unicast communication.

8.4.2.2 MIS PDU protected through EAP-generated MIS SA

An MIS security association (SA) may be established through an MIS service access authentication. An MIS SA is established for a pair of MISFs. It includes a ciphersuite used for the protection. A security association identifier is assigned by the PoS as a result of successful extensible authentication protocol (EAP) execution and communicated to the MN via an MIS_Auth request message with a Status indicating Success. Figure 32 shows a protected MIS PDU. The protection procedure is specified in 9.3.1.

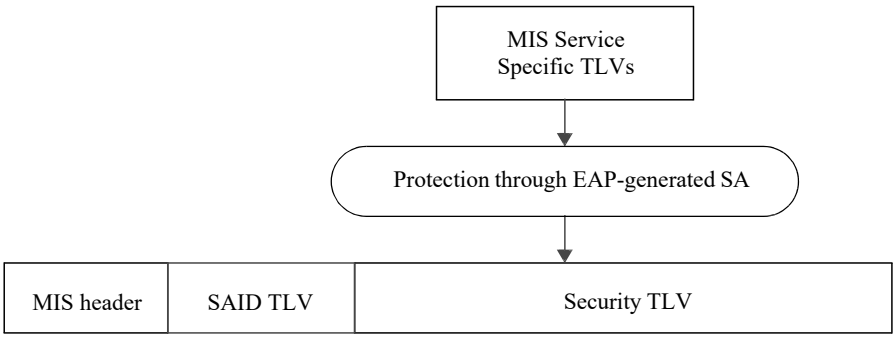


Figure 32—MIS PDU protected by an EAP-generated MIS SA

A signature TLV shall not be carried when MIS PDU is protected through EAP-generated MIS SA.

8.4.2.3 Protected MIS PDU upon transport address change

If the transport address of an MISF peer changes over the lifetime of a TLS session or the lifetime of an SA, the mapping between the sender's transport address and the MISF identifier shall be updated only if the MISF identifier is the same as that is currently bound to the security association identifier, and an MIS registration request or response message may be contained in the security TLV. The structure of a protected MIS PDU upon transport address change is shown in Figure 33.

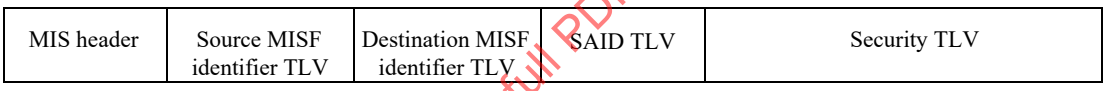


Figure 33—MIS PDU upon Transport Address Change

8.4.2.4 MIS PDU protected through Group key–generated MIS SA

When a GKB is used to distribute a master group key (MGK), the keys derived from MGK shall be used for a group MIS SA that is created for a group of MISFs to encrypt service specific TLVs of an MIS PDU. The group MIS SA is identified by a security association identifier assigned by the PoS and carried in a SAID TLV. For integrity protection, a signature TLV is carried in the MIS PDU. The signature TLV is allowed to be omitted if security against insiders is not required. Figure 34 shows a protected MIS PDU for GKB-generated MIS SA. The protection procedure is specified in 9.6.2.

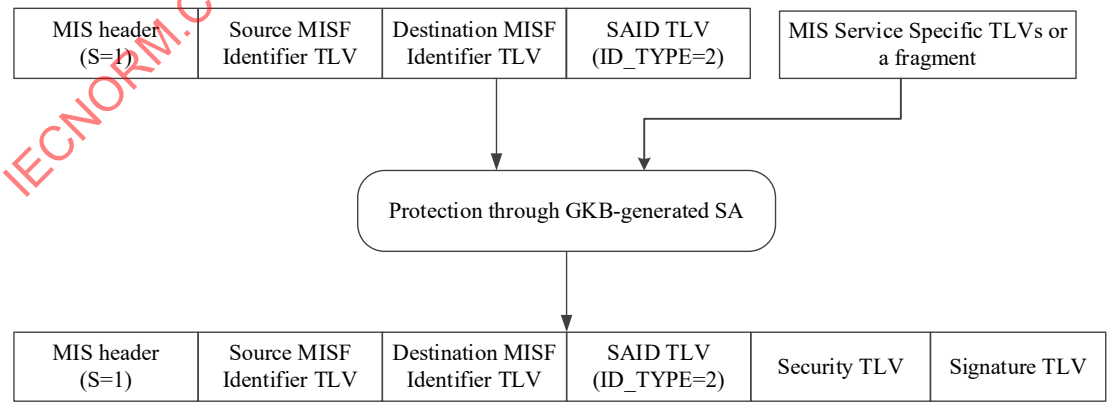


Figure 34—MIS PDU protected by a GKB-generated MIS SA with a signature TLV

8.4.2.5 MIS PDU protected by digital signature only

When an MIS PDU with the S bit set sent by a PoS is not encrypted, it is integrity protected by a digital signature. Figure 35 shows an MIS PDU protected by a digital signature only. The protection procedure is specified in 9.6.

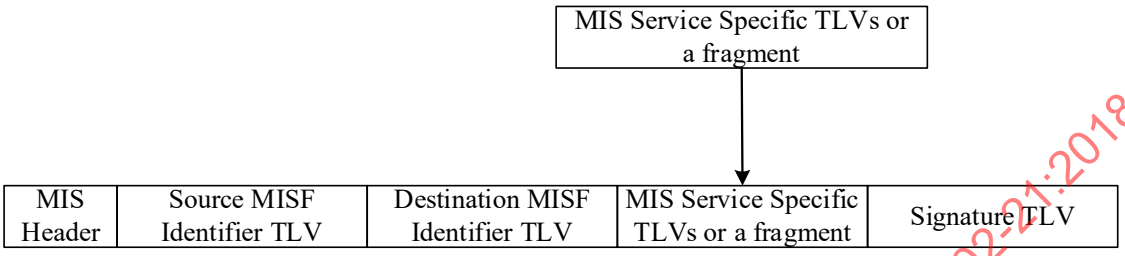


Figure 35—MIS PDU protected by digital signature only

8.4.3 Fragmentation and reassembly

8.4.3.1 General

The MIS fragmentation mechanism is defined using 'M' (More Fragment) and 'FN' (Fragment Number) fields of the MIS protocol header.

An MIS message is fragmented only when MIS message is sent natively over an L2 medium such as Ethernet. The message is fragmented when the message size exceeds aFragmentationThreshold. The size of each of the fragments is the same except the last one, which may be smaller. The maximum fragment size is defined as the maximum value of a FragmentationThreshold, which shall be equal to the Maximum Transmission Unit (MTU) (in octets) of the link layer that is on the path between two MISF nodes, minus securityOverhead octets, which is the maximum expansion for each protected MIS PDU. When there is no MIS SA, securityOverhead is set to '0'. The calculation of securityOverhead when there is an MIS SA is given in Annex J. When the MTU of the link layer between two MISF nodes is known, the maximum fragment size is set to the MTU (in octets) minus securityOverhead octets. The method of determining such an MTU is outside the scope of this standard. When the MTU of the link layer between two MISF nodes is unknown, the maximum fragment size is set to the minimum MTU of 1500 octets minus securityOverhead octets. When MIS message is sent using an L3 or higher layer transport, L3 takes care of any fragmentation issue, and the MIS protocol does not handle fragmentation in such cases.

Figure 36 shows the components of the fragmented MIS protocol frame. The MIS protocol payload carries a source MISF identifier TLV and a destination MISF identifier TLV followed by a fragment payload. Based on the fragment size, the fragment payload has a chance of being not aligned on a TLV boundary, i.e., TLVs other than the Source MISF identifier and destination MISF identifier TLVs should not be complete within the fragment payload. The fragment size may be smaller than the maximum fragment size and shall be large enough such that the number of fragments does not exceed 127.

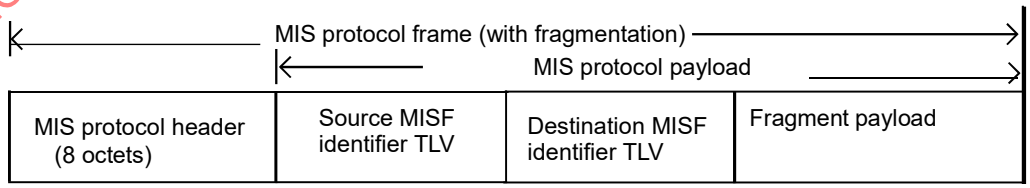


Figure 36—Fragmented MIS protocol frame format

When an MIS PDU is protected, the protection is applied to the fragment payload as shown in Figure 37.

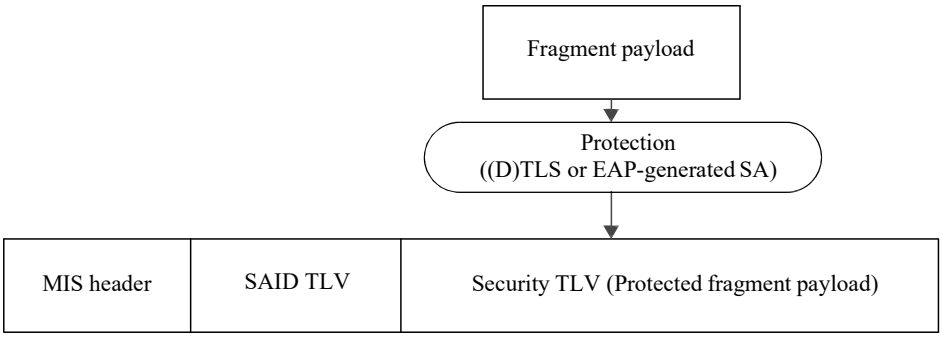


Figure 37—Protected fragmented MIS protocol frame format

8.4.3.2 Fragmentation

When an MIS message is fragmented, the fragmentation is performed within 'Transmit()' procedure in the MIS transaction protocol state machines. The MIS protocol header, the source MISF identifier TLV and destination MISF identifier TLV of the original message are copied to each fragment. When an MIS SA exists, the S bit in the header is set to '1' and an SAID TLV is included in each fragment. In this case, the source MISF identifier TLV and destination MISF identifier TLV of the original message are optional. However the 'variable payload length', 'more fragment', and 'fragment number' fields are updated accordingly for each fragment.

Variable payload length of each fragment indicates the number of octets in the MIS protocol payload of that fragment.

'More fragment' and 'fragment number' fields of each fragment are set according to the description in Table 22.

When data are to be transmitted, the number of octets in the fragment shall be determined by the fragment size and the number of octets in the multi-fragment message that have yet to be assigned to a fragment at the instant the fragment is constructed for the first time. Once a fragment is transmitted for the first time, its frame body content and length shall be fixed until it is successfully delivered to the destination MISF.

No retransmission by the MIS protocol (defined in 8.2) is performed for any single fragment of a multi-fragment message.

8.4.3.3 Reassembly

The destination MISF reassembles the received fragments into an original message. Reassembly is performed outside the MIS transaction state machines. 'MsgIn' and 'MsgInAvail' variables are set only after successful reassembly. An MISF shall be capable of receiving fragments of arbitrary length.

The following fields are used for reassembling fragments:

- S bit
- MIS message ID
- Transaction ID
- Source MISF identifier TLV
- Destination MISF identifier TLV
- SAID TLV (when source and destination MISF identifiers are not present)
- More fragment
- Fragment number

When any fragment of a multi-fragment message has arrived first, the destination MISF starts a timer referred to as ReassemblyTimer. If this ReassemblyTimer expires before all fragments have been received, the destination MISF discards those fragments that it has received. A duplicate fragment is discarded.

When S bit is set to '1', the fragment is protected. The protected fragment is verified for its integrity at the receiving end. If encryption is applied, it is decrypted to obtain the plaintext fragment. The security association identifier maps the fragment to a pair of source and destination MISF identifiers that are required for reassembly. The reassembly is performed after all the fragments are verified and decrypted.

An example of an original MIS message and fragmented MIS messages is shown in Annex J.

8.5 Message parameter TLV encoding

The following general TLV encoding shown in Figure 38 shall be used for all parameters in an MIS protocol message.

Type (1 octet)	Length (variable octets)	Value (variable octets)
Type of this parameter	Length of the *value* field of this parameter	Value of this parameter

Figure 38—Message parameter TLV encoding

Specifically, the *Type* field is one octet[14], and the *Length* shall be encoded with the rules described in 6.5.6.2.

Moreover, TLV *Type* values shall be unique within the MIS protocol. The TLV encoding starts at 1 and any subsequent values are assigned in ascending order (see Table K.2).

The TLV encoding of the vendor-specific TLV (type = 111) is shown in Figure 39.

Type (1 octet)	Length (variable octets)	Value (variable octets)	
111	Length of *value* field	OUI[a] or CID[a] (3 octets)	Vendor specific content

[a] Interested applicants should contact the IEEE Registration Authority, http://standards.ieee.org/regauth .

Figure 39—The TLV encoding for the vendor-specific TLV (Type = 111)

8.6 MIS protocol messages

The following subclauses specify different MIS protocol messages in TLV form. The shaded areas represent the MIS protocol header, while the unshaded areas represent the MIS protocol payload. The payload consists of a set of identifiers in TLV form.

The TLV Type assignment for each TLV can be found in Table K.2.

TLV type values ranging from 112 to 255 are reserved for experimental TLVs. These values are used by different implementations to evaluate the option of using TLVs not defined by the specification.

[14] The TLV *Type* field length is different than the Information Element *Type* length, which is four octets.

When a TLV type value is in the range of experimental TLVs and the data type of the TLV value is unknown or the TLV value is not in the range of valid values, the TLV should be ignored and the rest of the message should be processed. Also, experimental TLVs can be ignored, based on the MISF information that is communicating with another MISF with different experimental TLVs implementation.

All MIS messages carry a source MISF ID followed by a destination MISF ID as the first two TLVs of the MIS protocol payload part of the message. Zero length MISF ID can be used in MIS_Capability_Discover request and response messages as its destination MISF ID.

All "Optional" fields are optionally sent, but the receiver shall properly operate on them if present, i.e., these fields are mandatory in the implementation, but optional in their use.

On receipt of an MIS request message the MISF shall respond with a corresponding response message.

Any message received that has an invalid MIS header, or does not contain the source/destination MISF IDs, or has an unrecognizable or invalid MIS Message ID, shall be discarded without sending any indication to the source MIS node. Any undefined or unrecognizable TLVs in a received message shall be ignored by the receiver.

8.6.1 MIS messages for service management

8.6.1.1 MIS_Capability_Discover request

The corresponding MIS primitive of this message is defined in 7.4.1.1.

If a requesting MISF entity knows the destination MISF entity's MISF ID, the requesting MISF entity fills its destination MISF ID and sends this message to the peer MISF over the data plane, either L2 or L3.

If a requesting MISF entity does not know the destination MISF entity's MISF ID, the requesting MISF entity should fill its destination MISF ID with an MISF Broadcast ID to send this capability discover message.

MIS Header Fields (SID = 1, Opcode = 1, AID = 1)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkAddressList (optional) (Link address list TLV)
SupportedMISEventList (optional) (MIS event list TLV)
SupportedMISCommandList (optional) (MIS command list TLV)
SupportedISQueryTypeList (optional) (MIIS query type list TLV)
SupportedTransportList (optional) (Transport option list TLV)
SupportedSecurityCapList (optional) (Security capability TLV)
SupportedLinkActionsList (optional) (Link actions list TLV)

8.6.1.2 MIS_Capability_Discover response

The corresponding MIS primitive of this message is defined in 7.4.1.3. This message is sent in response to an MIS_Capability_Discover request message that was destined to a single or multicast MISF ID or an MISF Broadcast ID.

MIS Header Fields (SID = 1, Opcode = 2, AID = 1)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)
Link Address List (optional) (Link address list TLV)
SupportedMISEventList (optional) (MIS event list TLV)
SupportedMISCommandList (optional) (MIS command list TLV)
SupportedISQueryTypeList (optional) (MIIS query type list TLV)
SupportedTransportList (optional) (Transport option list TLV)
SupportedSecurityCapList (optional) (Security capability TLV)
SupportedLinkActionsList (optional) (Link actions list TLV)

8.6.1.3 MIS_Register request

The corresponding MIS primitive of this message is defined in 7.4.2.1.

This message is transmitted to the remote MISF to perform a registration or re-registration. The message shall contain the link identifier TLV or group link identifier TLV.

MIS Header Fields (SID = 1, Opcode = 1, AID = 2)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Optional) (Link identifier list TLV)
GroupLinkIdentifier (Optional) (Group link identifier TLV)
RequestCode (Register request code TLV)

8.6.1.4 MIS_Register response

The corresponding MIS primitive of this message is defined in 7.4.2.3.

This message is sent in response to a registration or re-registration request.

MIS Header Fields (SID = 1, Opcode = 2, AID = 2)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)
ValidTimeInterval (not included if Status does not indicate “Success”) (Valid time interval TLV)
MulticastCipherSuite (Optional) (Group Ciphersuite TLV)
Certificate (Optional) (Certificate TLV)

8.6.1.5 MIS_DeRegister request

The corresponding MIS primitive of this message is defined in 7.4.3.1.

This message is transmitted to the remote MISF to request a de-registration. There is no parameter for this message.

MIS Header Fields (SID = 1, Opcode = 1, AID = 3)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)

8.6.1.6 MIS_DeRegister response

The corresponding MIS primitive of this message is defined in 7.4.3.3.

This message is sent in response to a de-registration request.

MIS Header Fields (SID = 1, Opcode = 2, AID = 3)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)

8.6.1.7 MIS_Event_Subscribe request

The corresponding MIS primitive of this message is defined in 7.4.4.1.

This message is sent by a remote MISF (the subscriber) to subscribe to one or more event types from a particular event origination point. The message shall contain the link identifier TLV or group link identifier TLV.

MIS Header Fields (SID = 1, Opcode = 1, AID = 4)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Optional) (Link identifier TLV)
GroupLinkIdentifier (Optional) (Group link identifier TLV)
RequestedMISEventList (MIS event list TLV)
EventConfigurationInfoList (Optional) (Event configuration info list TLV)

8.6.1.8 MIS_Event_Subscribe response

The corresponding MIS primitive of this message is defined in 7.4.4.2.

The response indicates which of the event types were successfully subscribed.

MIS Header Fields (SID = 1, Opcode = 2, AID = 4)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)
LinkIdentifier (Link identifier TLV)
ResponseMISEventList (not included if Status does not indicate “Success”) (MIS event list TLV)

8.6.1.9 MIS_Event_Unsubscribe request

The corresponding MIS primitive of this message is defined in 7.4.5.1.

This message is sent by a remote MISF (the subscriber) to unsubscribe from a set of link-layer events. The message shall contain the link identifier TLV or group link identifier TLV.

MIS Header Fields (SID = 1, Opcode = 1, AID = 5)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Optional) (Link identifier TLV)
GroupLinkIdentifier (Optional) (Group link identifier TLV)
RequestedMISEventList (MIS event list TLV)

8.6.1.10 MIS_Event_Unsubscribe response

The corresponding MIS primitive of this message is defined in 7.4.5.2.

The response indicates which of the event types were successfully unsubscribed.

MIS Header Fields (SID = 1, Opcode = 2, AID = 5)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)
LinkIdentifier (Link identifier TLV)
ResponseMISEventList (not included if Status does not indicate “Success”) (MIS event list TLV)

8.6.1.11 MIS_Auth indication

This is used for an MISF to perform (D)TLS exchange with another MISF to establish or terminate a (D)TLS-generated MIS SA. It is also used to initiate an MIS service access authentication through EAP or ERP. In the former case, an AuthenticationContent shall be included to carry a TLS record of type handshake, change ciphersuite or alert message. In the latter case, this message is used in two different situations: a) when EAP execution is initiated by the MN; b) when ERP execution is initiated by the PoS. Only in case b), AuthenticationContent shall be included to carry an ERP message (ERP-Initiate/Re-auth-Start). This message shall not be used when EAP execution is initiated by a PoS or when ERP execution is initiated by an MN; an MIS_Auth request message shall be used instead.

MIS Header Fields (SID = 1, Opcode = 3, AID = 6)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
AuthenticationContent (optional) (Authentication TLV)

8.6.1.12 MIS_Auth request

This message is used for an MISF in either an MN or a PoS to send EAP or ERP messages in an MIS service authentication.

MIS Header Fields (SID = 1, Opcode = 1, AID = 6)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Security association ID (optional) (SAID TLV)
Nonce (optional) (Nonce TLV)
AuthenticationContent (optional) (Authentication TLV)
KeyLifeTime (optional) (KeyLifeTime TLV)
Status (optional) (STATUS TLV)
CipherSuite(optional) (Ciphersuite TLV)
AUTH (optional) (AUTH TLV)

8.6.1.13 MIS_Auth response

This message is used for an MISF in either an MN or a PoS to send EAP or ERP messages in an MIS service authentication.

MIS Header Fields (SID = 1, Opcode = 2, AID = 6)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Nonce (optional) (Nonce TLV)
AuthenticationContent (optional) (Authentication TLV)
KeyLifeTime (optional) (KeyLifeTime TLV)
Status (optional) (STATUS TLV)
CipherSuite(optional) (Ciphersuite TLV)
AUTH (optional) (AUTH TLV)

8.6.1.14 MIS_Termination_Auth request

This message is used for an MISF in a PoS to terminate an MIS SA.

MIS Header Fields (SID = 1, Opcode = 1, AID = 7)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)

8.6.1.15 MIS_Termination_Auth response

This message is used for an MISF in an MN to terminate an MIS SA.

MIS Header Fields (SID = 1, Opcode = 2, AID = 7)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)

8.6.1.16 MIS_Push_key request

This message is used for an MISF to communicate to another MISF to push a media-specific master session key or media-specific master session keys to a specific PoA or PoAs. The corresponding primitive is defined in 7.4.17.1.

MIS Header Fields (SID = 1, Opcode = 1, AID = 8)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkTupleIdentifierList (Link tuple identifier list TLV)

8.6.1.17 MIS_Push_key response

This message is used for an MISF to communicate to another MISF that a media-specific master session key or media-specific master session keys are installed in a specific PoA or PoAs. The corresponding primitive is defined in 7.4.17.3.

MIS Header Fields (SID = 1, Opcode = 2, AID = 8)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkTupleIdentifierList (Link tuple identifier list TLV)
Status (optional) (STATUS TLV)

8.6.1.18 MIS_LL_Auth request

This message is used for an MISF to carry link-layer frames to conduct an authentication. The corresponding primitive is defined in 7.4.18.1.

MIS Header Fields (SID = 1, Opcode = 1, AID = 9)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Link identifier TLV)
LLInformation (Link layer information TLV)

8.6.1.19 MIS_LL_Auth response

This message is used for an MISF to carry link-layer frames to conduct an authentication. The corresponding primitive is defined in 7.4.18.3.

MIS Header Fields (SID = 1, Opcode = 2, AID = 9)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Link identifier TLV)
LLInformation (Link layer information TLV)
Status (Status TLV)

8.6.1.20 MIS_Configuration_Update indication

The corresponding MIS primitive of this message is defined in 7.4.19.1.

This message is used by the MISF to change configuration of the MIS node(s) identified by the Destination Identifier.

The Destination Identifier is passed to the local MIS user as a TargetIdentifier in an MIS_Configuration_Update.indication.

MIS Header Fields (SID = 1, Opcode = 3, AID = 10)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
ConfigurationData (Configuration data TLV)

8.6.1.21 MIS_Configuration_Update request

The corresponding MIS primitive of this message is defined in 7.4.19.1.

This message is used by the MISF to change configuration of the MIS node(s) identified by the Destination Identifier.

The Destination Identifier is passed to the local MIS user as a TargetIdentifier in an MIS_Configuration_Update.indication.

MIS Header Fields (SID = 1, Opcode = 1, AID = 10)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
ConfigurationData (Configuration data TLV)

8.6.1.22 MIS_Configuration_Update response

The corresponding MIS primitive of this message is defined in 7.4.19.3.

This message is used by the MISF to inform the status of configuration update.

MIS Header Fields (SID = 1, Opcode = 2, AID = 10)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)

8.6.1.23 MIS_Pull_Group_Manipulate request

The corresponding MIS primitive of this message is defined in 7.4.20.1.

This message is used by the MISF to manipulate group membership of MIS node(s) identified by the Destination Identifier.

MIS Header Fields (SID = 1, Opcode = 1, AID = 11)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
TargetIdentifier (Group identifier TLV)
GroupAction (Group action TLV)

8.6.1.24 MIS_Pull_Group_Manipulate response

The corresponding MIS primitive of this message is defined in 7.4.20.3.

This message is used by the MISF to supply the group status of MIS node(s) identified by the Source Identifier.

MIS Header Fields (SID = 1, Opcode = 2, AID = 11)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
TargetIdentifier (Group identifier TLV)
SequenceNumber (conditional)[a] (Sequence number TLV)
TransportAddress (optional) (Transport address TLV)
SubgroupRange (optional) (Subgroup range TLV)
UserSpecificData (optional) (Aux data TLV)
CompleteSubtree (optional) (Complete subtree TLV)
ComplementSubtreeFlag (optional)[b] (Complement subtree flag TLV)
GroupKeyData (optional) (Group key data TLV)
GroupStatus (Group status TLV)
VerifyGroupCode (optional) (Verify group code TLV)
SecurityAssociationID (optional) (SAID notification TLV)

[a] This parameter is carried for CCM counter synchronization purpose when GroupKeyData is carried.
[b] If ComplementSubtreeFlag is not present, it means ComplementSubtreeFlag = 0.

8.6.1.25 MIS_Push_Group_Manipulate indication

The corresponding MIS primitive of this message is defined in 7.4.21.1.

This message is used by the MISF to manipulate group membership of MIS node(s) identified by the Destination Identifier.

MIS Header Fields (SID = 1, Opcode = 3, AID = 12)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
TargetIdentifier (Group identifier TLV)
SequenceNumber (conditional)[a] (Sequence number TLV)
TransportAddress (optional) (Transport address TLV)
SubgroupRange (optional) (Subgroup range TLV)
UserSpecificData (optional) (Aux data TLV)
CompleteSubtree (optional) (Complete subtree TLV)
ComplementSubtreeFlag (optional)[b] (Complement subtree flag TLV)
GroupKeyData (optional) (Group key data TLV)
VerifyGroupCode (optional) (Verify group code TLV)
SecurityAssociationID (optional) (SAID notification TLV)

[a] This parameter is carried for CCM counter synchronization purpose when GroupKeyData is carried.
[b] If ComplementSubtreeFlag is not present, it means ComplementSubtreeFlag = 0.

8.6.1.26 MIS_Push_Group_Manipulate request

The corresponding MIS primitive of this message is defined in 7.4.21.1.

This message is used by the MISF to manipulate group membership of MIS node(s) identified by the Destination Identifier.

MIS Header Fields (SID = 1, Opcode = 1, AID = 12)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
TargetIdentifier (Group identifier TLV)
SequenceNumber (conditional)[a] (Sequence number TLV)
TransportAddress (optional) (Transport address TLV)
SubgroupRange (optional) (Subgroup range TLV)
UserSpecificData (optional) (Aux data TLV)
CompleteSubtree (optional) (Complete subtree TLV)
ComplementSubtreeFlag (optional)[b] (Complement subtree flag TLV)
GroupKeyData (optional) (Group key data TLV)
VerifyGroupCode (optional) (Verify group code TLV)
SecurityAssociationID (optional) (SAID notification TLV)

[a] This parameter is carried for CCM counter synchronization purpose when GroupKeyData is carried.
[b] If ComplementSubtreeFlag is not present, it means ComplementSubtreeFlag = 0.

8.6.1.27 MIS_Push_Group_Manipulate response

The corresponding MIS primitive of this message is defined in 7.4.21.3.

This message is used by the MISF to inform group status of MIS node(s) identified by the Source Identifier.

MIS Header Fields (SID = 1, Opcode = 2, AID = 12)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
TargetIdentifier (Group identifier TLV)
GroupStatus (Group status TLV)

8.6.1.28 MIS_Pull_Certificate request

The corresponding MIS primitive of this message is defined in 7.4.22.1.

This message is used by the MISF to request a PoS certificate from the PoS identified by the Destination Identifier.

MIS Header Fields (SID = 1, Opcode = 1, AID = 13)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)

8.6.1.29 MIS_Pull_Certificate response

The corresponding MIS primitive of this message is defined in 7.4.22.3.

This message is used by the MISF to deliver a PoS certificate from a PoS to the sender of an MIS_Pull_Certificate request message.

MIS Header Fields (SID = 1, Opcode = 2, AID = 13)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Certificate (Certificate TLV)

8.6.1.30 MIS_Push_Certificate indication

The corresponding MIS primitive of this message is defined in 7.4.23.1.

This message is used by the MISF to deliver a credential encrypted by the leaf key that the MIS node identified by the Destination Identifier holds to the MIS node.

MIS Header Fields (SID = 1, Opcode = 3, AID = 14)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Certificate (Certificate TLV)

8.6.1.31 MIS_Push_Certificate request

The corresponding MIS primitive of this message is defined in 7.4.23.1.

This message is used by the MISF to deliver a credential encrypted by the leaf key that the MIS node identified by the Destination Identifier holds to the MIS node.

MIS Header Fields (SID = 1, Opcode = 1, AID = 14)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Certificate (Certificate TLV)

8.6.1.32 MIS_Push_Certificate response

The corresponding MIS primitive of this message is defined in 7.4.23.3.

This message is used by the MISF to acknowledge receipt of a credential from a PoS.

MIS Header Fields (SID = 1, Opcode = 2, AID = 14)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
CertificateSerialNumber (Certificate serial number info TLV)
CertificateStatus (Certificate status TLV)

8.6.1.33 MIS_Revoke_Certificate indication

The corresponding MIS primitive of this message is defined in 7.4.24.1.

This message is used by the MISF to revoke a credential.

MIS Header Fields (SID = 1, Opcode = 3, AID = 15)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
(Optional) CertificateSerialNumber (Certificate serial number info TLV)
CertificateRevocation (Certificate revocation signature TLV)
(Optional) IssuerName (Issuer name TLV)

8.6.1.34 MIS_Revoke_Certificate request

The corresponding MIS primitive of this message is defined in 7.4.24.1.

This message is used by the MISF to revoke a credential.

MIS Header Fields (SID = 1, Opcode = 1, AID = 15)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
(Optional) CertificateSerialNumber (Certificate serial number info TLV)
CertificateRevocation (Certificate revocation signature TLV)
(Optional) IssuerName (Issuer name TLV)

8.6.1.35 MIS_Revoke_Certificate response

The corresponding MIS primitive of this message is defined in 7.4.24.3.

This message is used by the MISF to acknowledge receipt of a credential revocation request from a PoS.

MIS Header Fields (SID = 1, Opcode = 2, AID = 15)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
CertificateStatus (Certificate status TLV)

8.6.2 MIS messages for event services

8.6.2.1 MIS_Link_Detected indication

The corresponding MIS primitive of this message is defined in 7.4.6.

This message is transmitted to the remote MISF when a new link has been detected.

MIS Header Fields (SID = 2, Opcode = 3, AID = 1)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkDetectedInfoList (Link detected info list TLV)

8.6.2.2 MIS_Link_Up indication

The corresponding MIS primitive of this message is defined in 7.4.7.

This notification is delivered from an MISF, when present in the PoA, to an MISF in the network when a layer 2 connection is successfully established with an MN.

MIS Header Fields (SID = 2, Opcode = 3, AID = 2)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Link identifier TLV)
OldAccessRouter (optional) (Old access router TLV)
NewAccessRouter (optional) (New access router TLV)
IPRenewalFlag (optional) (IP renewal flag TLV)

8.6.2.3 MIS_Link_Down indication

The corresponding MIS primitive of this message is defined in 7.4.8.

This notification is delivered from an MISF, when present in the PoA, to an MISF in the network when a layer 2 connection with an MN is disconnected due to a certain reason.

MIS Header Fields (SID = 2, Opcode = 3, AID = 3)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Link identifier TLV)
OldAccessRouter (optional) (Old access router TLV)
ReasonCode (Link down reason code TLV)

8.6.2.4 MIS_Link_Parameters_Report indication

The corresponding MIS primitive of this message is defined in 7.4.9.

This message indicates changes in link conditions that have crossed pre-configured threshold levels. A pre-configured threshold level is set by the MIS_Link_Configure_Thresholds request message.

MIS Header Fields (SID = 2, Opcode = 3, AID = 5)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Link identifier TLV)
LinkParameterReportList (Link parameter report list TLV)

8.6.2.5 MIS_Link_Going_Down indication

The corresponding MIS primitive of this message is defined in 7.4.10.

This message is transmitted to the remote MISF when a layer 2 connectivity is expected (predicted) to go down within a certain time interval.

MIS Header Fields (SID = 2, Opcode = 3, AID = 6)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (Link identifier TLV)
TimeInterval (Time interval TLV)
LinkGoingDownReason (Link going down reason TLV)

8.6.3 MIS messages for command service

8.6.3.1 MIS_Link_Get_Parameters request

The corresponding MIS primitive of this message is defined in 7.4.12.2.

This message is used to discover the status of currently available links. The message shall contain the link identifier list TLV or group link identifier TLV.

MIS Header Fields (SID = 3, Opcode = 1, AID = 1)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
DeviceStatesRequest (Optional) (Device states request TLV)
LinkIdentifier(Optional) (Link identifier list TLV)
GroupLinkIdentifier (Optional) (Group link identifier TLV)
GetStatusRequestSet (Get status request set TLV)

8.6.3.2 MIS_Link_Get_Parameters response

The corresponding MIS primitive of this message is defined in 7.4.12.3.

This message is used by an MISF to report the status of currently available links.

MIS Header Fields (SID = 3, Opcode = 2, AID = 1)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)
DeviceStatesResponseList (optional) (not included if Status does not indicate “Success”) (Device states response list TLV)
GetStatusResponseList (not included if Status does not indicate “Success”) (Get status response list TLV)

8.6.3.3 MIS_Link_Configure_Thresholds indication

The corresponding MIS primitive of this message is defined in 7.4.13.2.

This message is used to configure thresholds of the lower layer link when an MISF Group ID is used as Destination Identifier. The message shall contain the link identifier TLV or group link identifier TLV.

MIS Header Fields (SID = 3, Opcode = 3, AID = 2)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (optional) (Link identifier TLV)
GroupLinkIdentifier (optional) (Multicast link identifier TLV)
ConfigureRequestList (Configure request list TLV)

8.6.3.4 MIS_Link_Configure_Thresholds request

The corresponding MIS primitive of this message is defined in 7.4.13.2.

This message is used to configure thresholds of the lower layer link. The message shall contain the link identifier TLV or group link identifier TLV.

MIS Header Fields (SID = 3, Opcode = 1, AID = 2)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkIdentifier (optional) (Link identifier TLV)
GroupLinkIdentifier (optional) (Multicast link identifier TLV)
ConfigureRequestList (Configure request list TLV)

8.6.3.5 MIS_Link_Configure_Thresholds response

The corresponding MIS primitive of this message is defined in 7.4.13.3.

This message returns the status of a thresholds configuration request. The MISF generating this message generates MIS_Link_Parameters_Report indication message when the configured threshold is crossed.

MIS Header Fields (SID = 3, Opcode = 2, AID = 2)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)
LinkIdentifier (Link identifier TLV)
ConfigureResponseList (not included if Status does not indicate “Success”) (Configure response list TLV)

8.6.3.6 MIS_Link_Actions indication

The corresponding MIS primitive of this message is defined in 7.4.14.1.

This message is used to control the behavior of a set of lower layer links when an MISF Group ID is used as Destination Identifier. The message shall contain the link identifier TLV or group link identifier TLV.

MIS Header Fields (SID = 3, Opcode = 3, AID = 3)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkActionsList (optional) (Link actions list TLV)
GroupLinkActionsList (optional) (Group link actions list TLV)

8.6.3.7 MIS_Link_Actions request

The corresponding MIS primitive of this message is defined in 7.4.14.1.

This message is used to control the behavior of a set of lower layer links.

MIS Header Fields (SID = 3, Opcode = 1, AID = 3)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
LinkActionsList (optional) (Link actions list TLV)
GroupLinkActionsList (optional) (Group link actions list TLV)

8.6.3.8 MIS_Link_Actions response

The corresponding MIS primitive of this message is defined in 7.4.14.2.

This message returns the result of an MIS_Link_Actions request.

MIS Header Fields (SID = 3, Opcode = 2, AID = 3)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)
LinkActionsResultList (not included if Status does not indicate "Success") (Link actions result list TLV)

8.6.4 MIS message for information service

MIS Information Service uses only the following messages: MIS_Get_Information request, MIS_Get_Information response, or MIS_Push_Information. Due to the need to support different query types and the need for flexibility to customize the query and response, the parameters and their usage in these messages are substantially different from other MIS message parameters, and are therefore separately defined in the following subclauses.

8.6.4.1 MIS_Get_Information request

The corresponding MIS primitive of this message is defined in 7.4.15.1.

This message is used by an MISF to retrieve a set of Information Elements provided by the information service. A single MIS_Get_Information request message carries only one query list. However, there can be multiple queries in that list in the order of the most preferred query first.

MIS Header Fields (SID = 4, Opcode = 1, AID = 1)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
InfoQueryBinaryDataList (optional) (Info query binary data list TLV)
InfoQueryRDFDataList (optional) (Info query RDF data list TLV)
InfoQueryRDFSchemaURL (optional) (Info query RDF schema URL TLV)
InfoQueryRDFSchemaList (optional) (Info query RDF schema list TLV)
MaxResponseSize (optional) (Max response size TLV)
QuerierNetworkType (optional) (Network type TLV)
UnauthenticatedInformationRequest (Unauthenticated information request TLV)

8.6.4.2 MIS_Get_Information response

The corresponding MIS primitive of this message is defined in 7.4.15.3.

This is used as a response to the MIS_Get_Information request message. The total response message size shall not exceed the value indicated in the max response size TLV of corresponding MIS_Get_Information request message. The order of the query response shall be in the same order as the query requests.

MIS Header Fields (SID = 4, Opcode = 2, AID = 1)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
Status (Status TLV)
InfoResponseBinaryDataList (optional) (Info response binary data list TLV)
InfoResponseRDFDataList (optional) (Info response RDF data list TLV)
InfoResponseRDFSchemaURLList (optional) (Info response RDF schema URL list TLV)
InfoResponseRDFSchemaList (optional) (Info response RDF schema list TLV)

8.6.4.3 MIS_Push_Information indication

The corresponding MIS primitive of this message is defined in 7.4.16.1.

This is an indication to push operator policies or other network information to the MN.

MIS Header Fields (SID = 4, Opcode = 3, AID = 1)
Source Identifier = sending MISF ID (Source MISF ID TLV)
Destination Identifier = receiving MISF ID (Destination MISF ID TLV)
InfoResponseBinaryDataList (optional) (Info response binary data list TLV)
InfoResponseRDFDataList (optional) (Info response RDF data list TLV)
InfoResponseRDFSchemaURLList (optional) (Info response RDF schema URL list TLV)
InfoResponseRDFSchemaList (optional) (Info response RDF schema list TLV)

9. MIS protocol protection

This clause specifies options and mechanisms to protect remote messages in the media independent service protocol. The remote messages in the MIS protocol can be protected through the transport protocols at layer 2 or layer 3. The protection through the transport protocols are discussed in Annex N. This clause specifies the mechanisms to protect MIS PDUs at the MIS layer. These mechanisms apply protection to MIS PDUs without depending on transport protocols. They are called MIS specific protection mechanisms. To apply MIS specific protection mechanisms, a mobile node and a point of service (PoS) need to negotiate protection mechanisms and to establish cryptographic keys. MIS message protection shall be accomplished in either of two ways. The first is to use TLS or DTLS, and the second is to use EAP or ERP as an MIS service access authentication to establish MIS security associations (SAs). If MIS service access authentication is needed and an authentication server is available, then EAP based authentication and key establishment may be used for establishing an MIS SA. In situations where MIS service access authentication is not required and TLS credentials are available, or where MIS service access authentication is required and TLS credentials for access authentication are available at a PoS, then (D)TLS may be used for establishing an MIS SA.

9.1 Protection established through MIS (D)TLS

In this option, a mobile node, the client, and a PoS, the server, execute a TLS, specified in IETF RFC 5246, or DTLS, specified in IETF RFC 6347, to establish MIS protection. When the MIS protocol transport is reliable, TLS is used. Otherwise, DTLS is used. In the rest of this standard, (D)TLS is used to denote TLS or DTLS. In a (D)TLS handshake, the mutual authentication is executed through either a pre-shared key or a public key certified by a trusted third party such as a certificate authority (CA). It should be noted that all certificates are required to be validated. The TLS certificate used by the PoS is expected to be provided to the mobile node in a secure manner, e.g., during provisioning process. In this option, the authentication may or may not be related to access control. It can be an access authentication for MIS service if a PoS holds service credentials for the mobile nodes.

After the handshake, a (D)TLS session is established. In this case, the TLS master key and the keys derived from the master key, all the TLS parameters, and TLS ciphersuite negotiated in the TLS handshake form an MIS SA. The (D)TLS security association identifier is carried in each message in the SAID TLV.

In a (D)TLS session, an MIS message is first protected as application data. Then the (D)TLS record is transported by MIS protocol by security TLV.

For a (D)TLS-generated MIS SA, it can be terminated through (D)TLS session termination using an MIS_Auth indication message.

9.2 Key establishment through an MIS service access authentication

If MIS service is subscription based and provided by a service provider, then an MIS service access authentication should be needed to authorize the service to a mobile node. In this case, a PoS should obtain a master session key through service access authentication and an MIS security associations can be established through the master session key between the MN and the PoS.

9.2.1 MIS service access authentication

In this standard, it is assumed that EAP (IETF RFC 3748) or EAP re-authentication (ERP) (IETF RFC 6696) is used as the authentication protocol with an MN as the peer and a PoS as the authenticator. An EAP server may be used as a backend server.

For the interface between an MN and a PoS, the MIS protocol is acting as an EAP lower layer. That is, at the MN, an EAP message is generated at the MISF. When it reaches the PoS, the MISF in the PoS processes it. For an EAP message from the PoS to the MN, the MISF in the PoS generates the message. At the MN, the EAP message is passed to the MISF to process. The protocol stack is illustrated in Figure 40, where it is assumed that an EAP server is employed. After a successful authentication, a master session key (MSK) is exported to the lower layer, that is, MIS layer. An MSK is used to further derive MIS message protection keys.

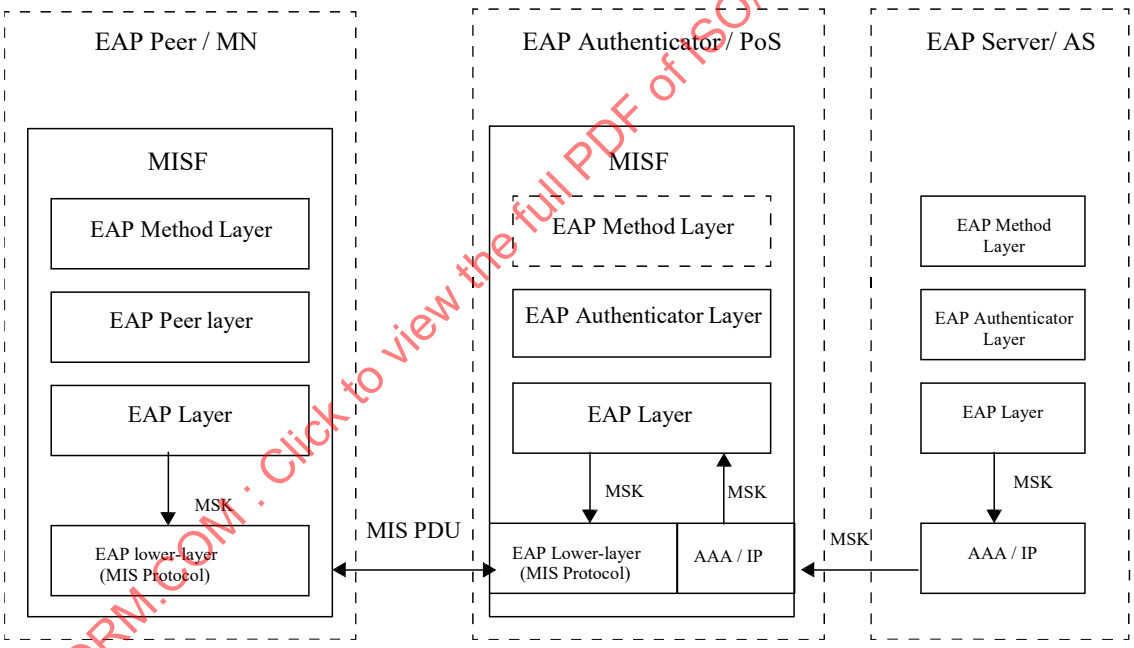


Figure 40—Protocol stack of service access authentication (with an EAP server)

The authentication is divided into the following phases:

a) *Capability Discovery Phase*. In this phase, both the MN and the PoS exchange unprotected MIS messages for an MN to discover the services that a PoS provides.

b) *MIS Service Access Authentication Phase*. Before starting the MIS access authentication, the MN and the PoS perform a negotiation in order to agree on a ciphersuite and other useful parameters to be used in the authentication and MIS message protection. The negotiation is initiated either by the MN or by the PoS. Once the negotiation is completed, the MN (acting as the EAP peer) authenticates against the PoS (acting as an EAP authenticator). To achieve this, EAP is transported by MIS protocol between the MN and the PoS. In order to carry out the authentication, the PoS has a choice of using a

backend authentication server (acting as an EAP server) to verify the MN's credentials. In this standard, it is assumed that the EAP methods employed can export keying material (i.e., MSK). Thus, after performing the authentication, keying material (i.e., MSK) is shared between the MN and the PoS. Specifically, the keying material is exported to MN's and PoS's lower layer (MIS layer) and used to protect the rest of the communication. The message protection mechanisms are specified in 9.3. The protected message format is specified in 8.4. In order to preserve the security of the exported keying material, the exported MSK is used as a root key to derive session keys which are used to protect the MIS PDUs. The key hierarchy is described in 9.2.2. Note that the authentication procedure could be based on an EAP re-authentication (ERP) in order to perform a fast authentication. In this case, an rMSK is used as the root key to derive the key hierarchy.

c) *Service Access Phase*. At this point, the MN is authenticated and authorized to use the MIS services, agreed and provided by the PoS. The MIS protocol is protected by using the keying material obtained in the MIS Service Access Authentication Phase. This phase is related to 9.2.2 for key derivation and 9.3 for protecting MIS protocol.

d) *Termination phase*. When the MN or the PoS desires to terminate the security association before the security association lifetime expires, either the MN or the PoS can request to terminate.

Figure 41 and Figure 42 illustrate the EAP execution when it is initiated by the MN and when it is initiated by the PoS respectively. In both figures, only the protocol interface between an EAP peer and an EAP authenticator is described. The interface with EAP server is not illustrated. MIS service access authentication messages are defined in 8.6.1.11, 8.6.1.12, and 8.6.1.13. Termination messages are defined in 8.6.1.14 and 8.6.1.15.

Similarly, Figure 43 illustrates an MN initiated ERP execution.

Figure 44 and Figure 45 show a PoS initiated ERP execution, where the ERP is initiated by sending an EAP Request/Identity as shown in Figure 44, or by sending an ERP-Initiate/Re-auth-Start as shown in Figure 45.

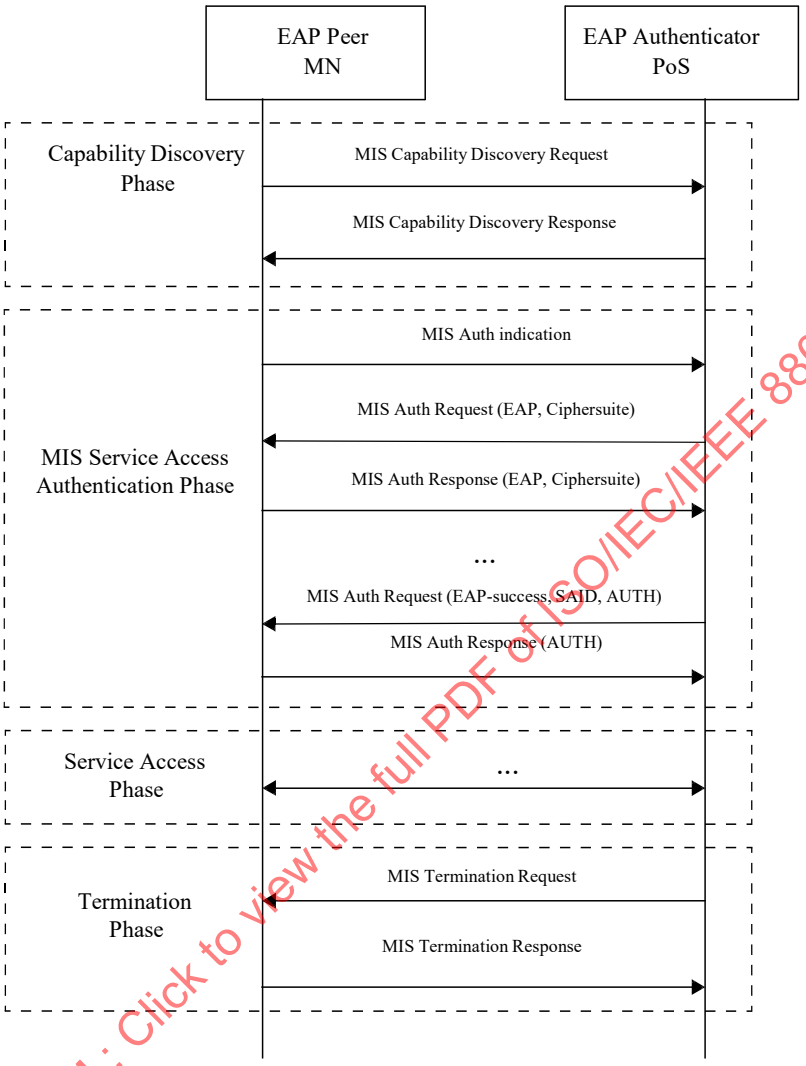


Figure 41—Main stages with MN initiated EAP execution

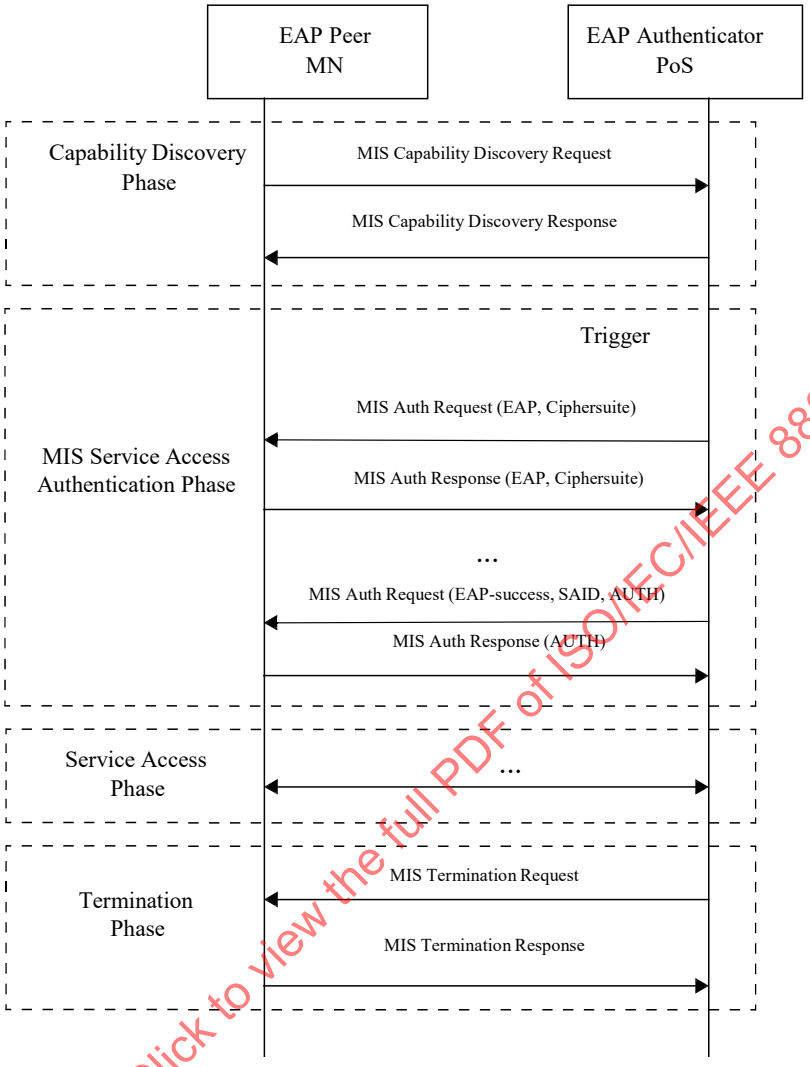


Figure 42—Main stages with PoS initiated EAP execution

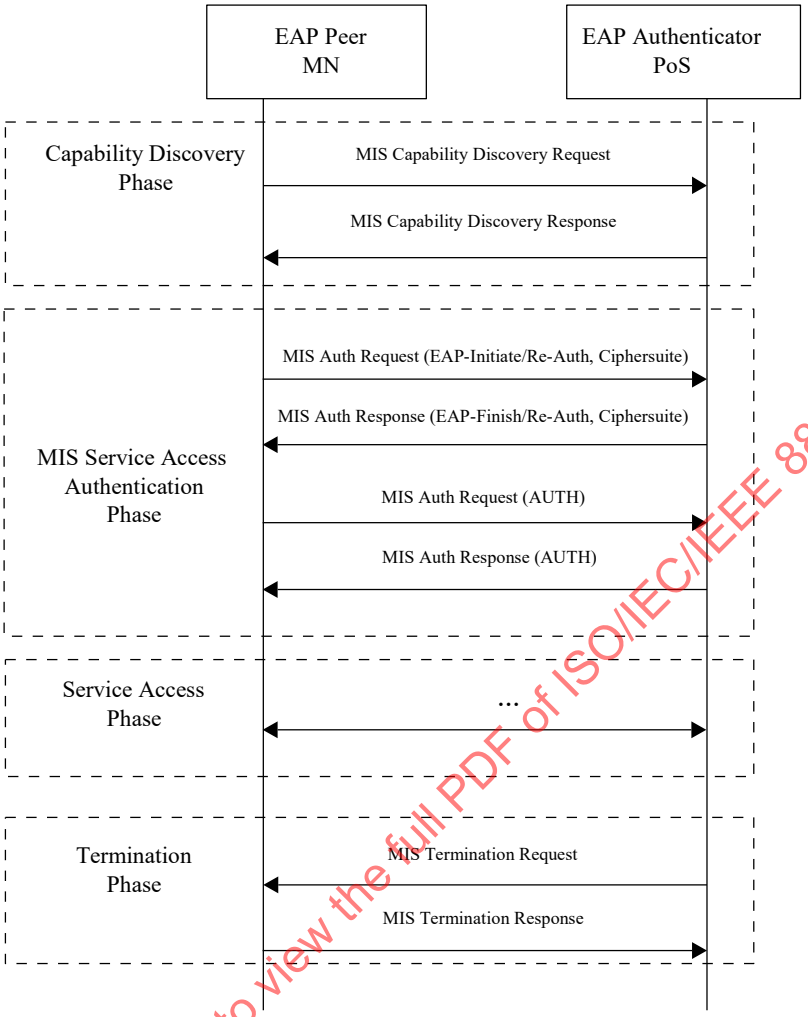


Figure 43—Main stages with MN initiated ERP execution

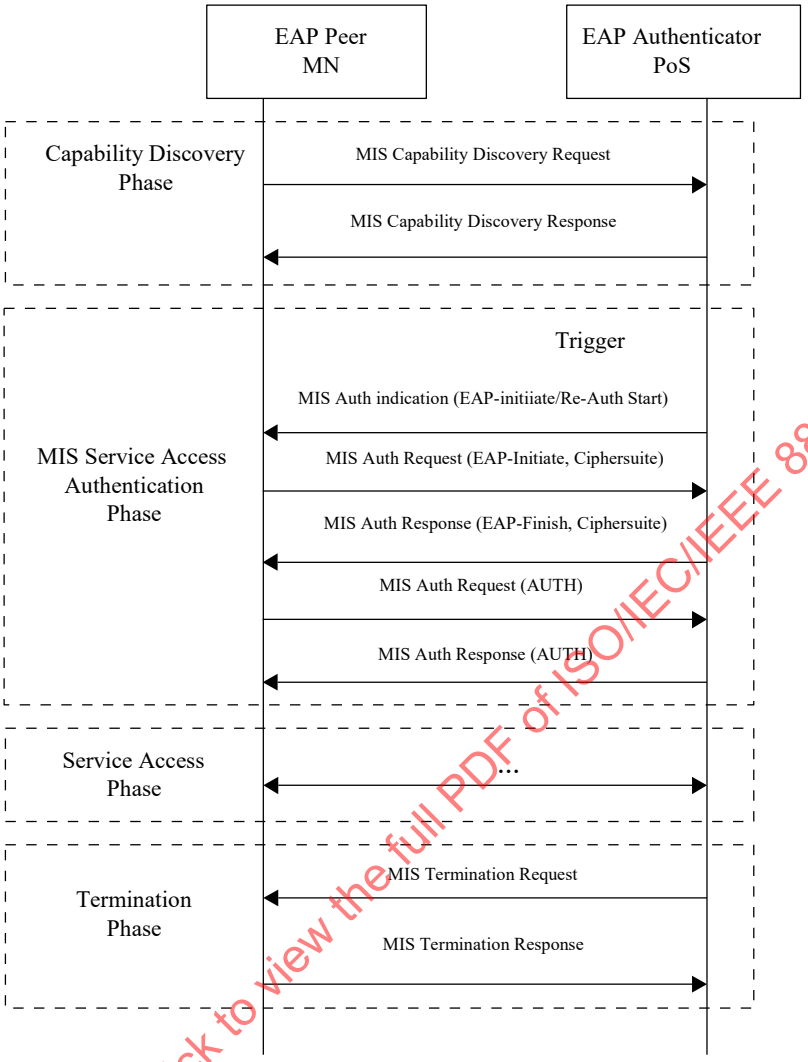


Figure 44—Main stages with PoS initiated ERP execution (1)

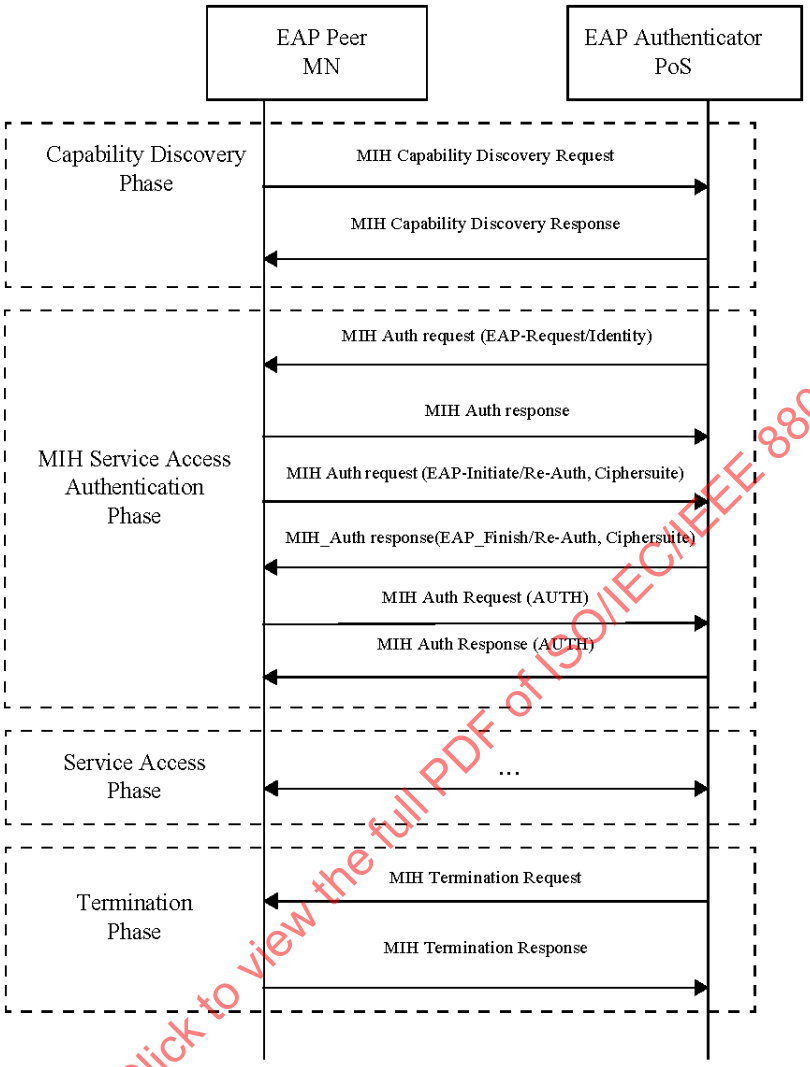


Figure 45—Main stages with PoS initiated ERP execution (2)

9.2.2 Key derivation and key hierarchy

Upon a successful MIS service access authentication, the authenticator (i.e., the PoS) obtains a master session key (MSK) or a re-authentication master session key (rMSK) via EAP to generate a KeyDerivationKey shared between the MN and the PoS.

The keys derived from KeyDerivationKey include a 128-bit authentication key (MIAK) used to generate a value AUTH; the session keys determined by the ciphersuite code *c* agreed upon between the MN and the serving PoS. If no ciphersuite code is specified by the MN, the default ciphersuite code is used as specified in 9.2.3. The session keys used for MIS message protection consist of an encryption key (MIEK) only, an integrity key (MIIK) only, or both an encryption key (MIEK) and an integrity key (MIIK). The concatenation of MIAK, MIEK, and MIIK is called the media independent session key (MISK). The length, *L*, of the MISK is specified in 9.2.3.

For the key derivation, the following notations and parameters are used:

— *K:* key derivation key. It is truncated from a master session key (MSK) or re-authentication MSK (rMSK). The length of *K* is determined by the pseudorandom function (PRF) used for key derivation. If HMAC-SHA-1 or HMAC-SHA-256 is used as a PRF, then the full MSK or rMSK is used as the key derivation key *K*. If CMAC-AES is used as a PRF, then the first 128 bits of MSK or rMSK are used as the key derivation key *K*.

— *L:* The binary length of derived keying material MISK. *L* is determined by the selected ciphersuite, which is specified in 9.2.3.

— *h:* The output binary length of PRF used in the key derivation. That is, *h* is the length of the block of the keying material derived by one PRF execution. Specifically, for HMAC-SHA-1, *h* = 160 bits; for HMAC-SHA-256, *h* = 256 bits; for CMAC-AES, *h* = 128 bits.

— *n:* The number of iterations of PRF in order to generate *L*-bits keying material.

— *Nonce-T* and *Nonce-N:* The nonces exchanged during the execution of service access authentication.

— *c:* The ciphersuite code is a one octet string specified for each ciphersuite. The code is defined in 9.2.3.

— *v:* The length of the binary representation of the counter and the length of keying material *L*. The default value for *v* is 32.

— "MISK": 0x4D49534B, ASCII code in hex for string "MISK."

— *[a]2:* Binary representation of integer *a* with a given length.

For a given PRF, the key derivation for MISK can be described in the following procedures:

Fixed input values: *h* and *v*.

Input: *K*, *Nonce-T*, *Nonce-N*, *L*, and ciphersuite code.

Process:

a) *n*:=[L/h];
b) If $n > 2^v - 1$, then indicate an error and stop.
c) Result (0) :=empty string.
d) For *i* = 1 to *n*, do
   i) *K(i) = PRF(K, "MISK" || [i]2 || Nonce-T || Nonce-N || c || [L]2).*
   ii) *Result(i) = Result (i − 1) || K(i).*
e) Return *Result (n)* and *MISK* is the leftmost *L* bits of *Result (n)*

The *MISK* is parsed in such a way that

MISK = MIAK || MIIK || MIEK

With the above procedure, a key hierarchy is derived as shown in Figure 46.

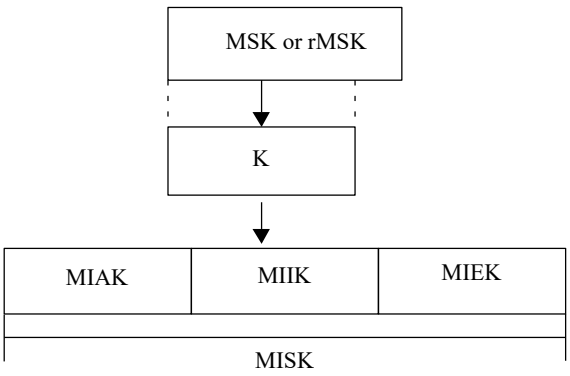


Figure 46—MIS Key Hierarchy

9.2.3 EAP-generated MIS security association

When an MIS SA is established through an EAP method with key establishment, the SA consists of the keys, the key derivation functions, and the ciphersuite. The key derivation functions, encryption algorithms, and integrity algorithms are specified in Table 24.

Table 24—Cryptographic algorithms

Encryption algorithm	**Description**
AES_CBC	Advanced encryption standard (AES) cipher block chaining (CBC) mode ([NIST SP 800-38A])
NULL	No encryption is applied
Integrity algorithm	**Description**
HMAC_SHA1_96	HMAC-SHA1 with 96 bits MAC ([FIPS 198])
AES_CMAC	AES CMAC mode with 128 bits MAC ([NIST SP 800-38B])
Authenticated encryption	**Description**
AES_CCM	AES-CCM mode ([NIST SP 800-38C])
PRF used for key derivation function	**Description**
PRF_CMAC_AES	AES CMAC as PRF in counter mode ([NIST SP 800-108])
PRF_HMAC_SHA1	HMAC-SHA1 as PRF in counter mode ([NIST SP 800-108])
PRF_HMAC_SHA256	HMAC-SHA256 as PRF in counter mode ([NIST SP 800-108])

The ciphersuites and key length are defined in Table 25.

Table 25—Ciphersuites

Code	Encryption algorithm	Integrity algorithm	MISK length (L)
00000010	AES_CBC	HMAC_SHA1_96	384
00000100	NULL	HMAC_SHA1_96	256
00000101	NULL	AES_CMAC	256
00000110	AES_CCM		256

A default ciphersuite is defined as AES_CCM. The default PRF is defined as PRF_CMAC_AES. The protection mechanisms for MIS messages are defined in 9.3.

9.2.4 Termination

A termination phase is defined as a mechanism to allow either an MN or a PoS to release the resource such as keys, authorized service access, etc. obtained through a service access authentication. Termination shall take place by either of two mechanisms:

a) Termination messages: These messages allow one party to explicitly inform another party the current authentication status is terminated. This option is supported by MIS_Termination_Auth messages defined in 8.6.1.14 and 8.6.1.15.

b) State timeout: A lifetime is defined for an MIS SA. After the time period defined by the lifetime, the MIS SA is terminated. The lifetime of the SA shall be no longer than the MSK or rMSK lifetime, and communicated to the MIS node acting as the EAP peer by the lifetime TLV included in MIS_Auth request and MIS_Auth response messages defined in 8.6.1.12 and 8.6.1.13.

9.3 MIS message protection mechanisms for EAP-generated SAs

9.3.1 MIS_Auth message protection

The MIS_Auth messages are not protected using the MIS SA established after a successful media independent service access authentication. MIS_Auth messages are integrity protected by including an AUTH TLV generated using MIAK derived from the MSK or rMSK, as described in 9.2.2, with a PRF. The AUTH TLV value is generated as follows:

AUTH TLV value = PRF(K, “AUTH-TLV” | MIS_Auth message| MNCiphersuite | PoSCiphersuite)

where

K	is MIAK
“AUTH-TLV”	is 0x415554482D544C56, ASCII code in hex for string “AUTH-TLV”
MIS_Auth message	is an MIS_Auth message in which AUTH TLV filled with 0s
MNCiphersuite	is ciphersuite TLV sent by the MN
PoSCiphersuite	is ciphersuite TLV sent by the PoS

PRF function is one of the following as negotiated:

a) PRF_CMAC_AES
b) PRF_HMAC_SHA1
c) PRF_HMAC_SHA256

The PRF output length shall be truncated to 128 bits. If the PRF output length is more than 128 bits, the 128 leftmost bits of the output shall be used as the AUTH TLV value.

9.3.2 MIS PDU protection procedure

Depending on the selected ciphersuite, an MIS PDU may be encrypted, integrity protected, or protected in both aspects. Correspondingly, an SA identifies either an encryption key (MIEK), an integrity key (MIIK), or both MIEK and MIIK. When both encryption and integrity protection are applied, they are accomplished by two algorithms such as AES in CBC mode and HMAC-SHA1_96 or by one authenticated encryption algorithm such as AES in CCM mode.

The portion to be protected in an MIS message includes the MIS service specific TLVs. That is, the source MISF identifier TLV and the destination MISF identifier TLV are not protected. The protection is applied based on an SA. An SAID TLV is carried in place of source and destination MISF identifier TLVs except for the case of transport address change where both an SAID TLV and source and destination MISF identifier TLVs are carried as described in 8.4.2. An example procedure is illustrated in Figure 47.

When fragmentation is applied to an MIS PDU, then instead of service specific TLVs, the data to be protected comprise a fragment payload. The values in the header fields M (more fragment) and FN (fragment number) are the same after the protection.

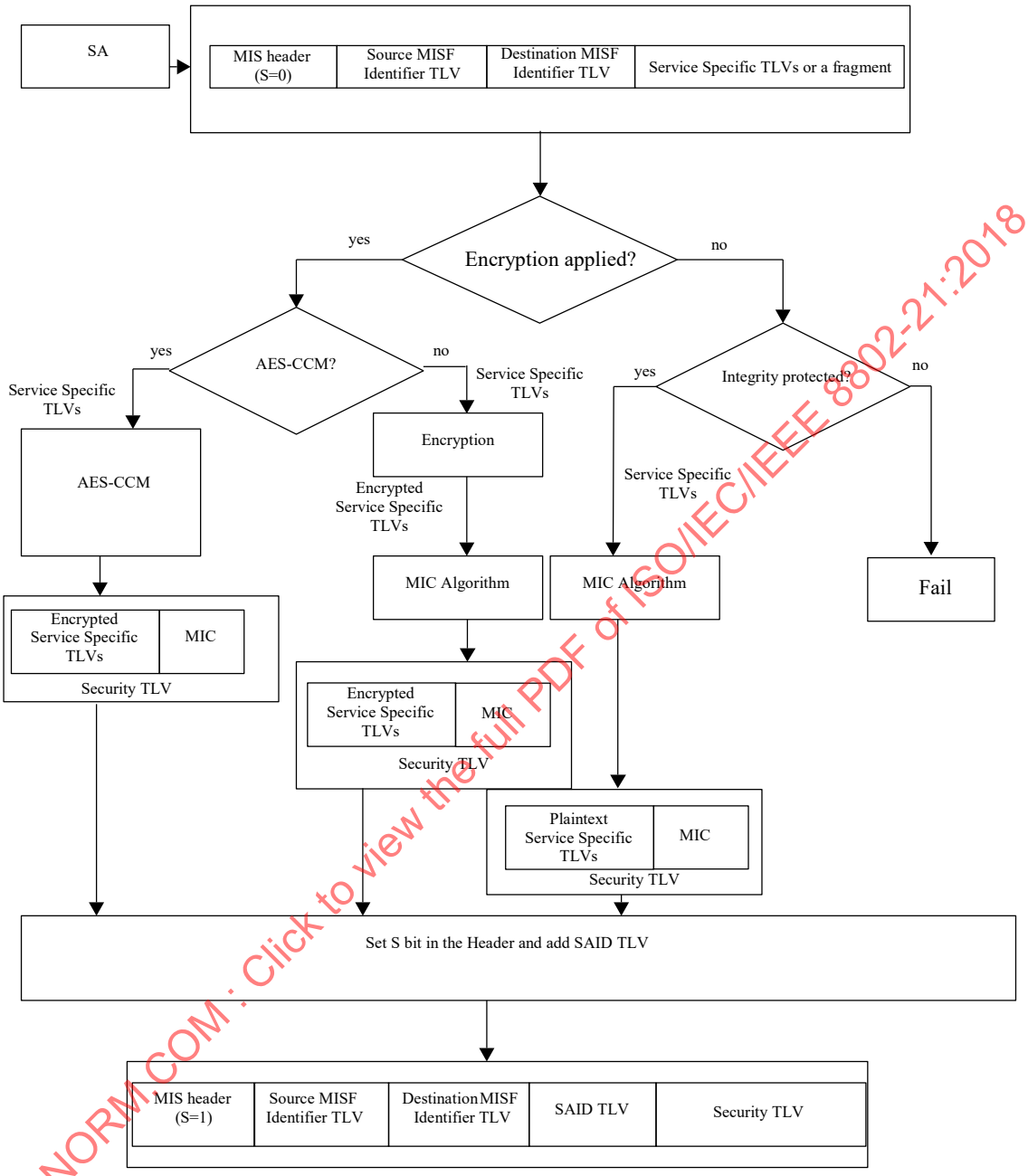


Figure 47—MIS PDU protection procedure

9.3.3 MIS PDU protection by AES-CCM

AES in CCM mode as specified in NIST SP 800-38C shall be the default ciphersuite. The parameters used in AES-CCM, the nonce construction, the operational procedures, and the security TLV under AES-CCM protection shall be set according to the rules given in 9.3.3.1 through 9.3.3.3.

9.3.3.1 AES-CCM parameters

For AES-CCM the following parameter values shall be set:

a) *t:* The length of MIC is 12 octets (96 bits).
b) *n:* The length of the nonce *N* is 13 octets (104 bits).
c) *q:* The length of the binary representation of the octet length of the data to be encrypted is 2 octets (16 bits).

9.3.3.2 Construct AES-CCM nonce

AES-CCM uses a nonce to construct an initialization vector and also the counter. CCM requires a unique nonce value for each MIS message protected by a given MIEK. In this standard, the nonce is 13 octets and consists of the following three portions.

a) Transaction ID (12 bits, from the MIS header) plus 4 reserved bits (set to '0');
b) Sequence number (10 octets, denoted as *SN0*, *SN1*, ..., *SN9*) starting from all zeroes; and
c) FN (7 bits, from the MIS header) plus 1 reserved bit (set to '0').

The nonce construction is illustrated in Figure 48.

Transaction ID (12)	Resv (4)	*SN0*, *SN1* ..., *SN9* (80)	FN (7)	Resv (1)

Figure 48—AES-CCM nonce construction

The *SN* is increased by a positive number for each MIS PDU. The SN shall never be repeated for a series of encrypted MIS PDUs using the same MIEK. For a given SA, each MISF keeps an SN, which is the highest positive value for a given MIEK.

9.3.3.3 Operational procedures in AES-CCM

9.3.3.3.1 Encapsulation

For a given SA, the prerequisites for AES-CCM encapsulation includes an encryption key MIEK, an AES block cipher encryption block, and the values of parameters *t*, *n*, and *q*. The plaintext, *P,* to be encrypted and authenticated is formed by concatenating all the service specific TLVs as presented in MIS PDU with the padding. In this standard, the associated data, A, is null. The data, *P,* is partitioned with necessary padding to 16-octet blocks *B1*, *B2*, ..., *Br* as specified in SP 800-38C. The octet block, *B0,* is an initialization vector and formed with 1-octet flags, 13-octet nonce *N*, and 2-octet integer *Q*, where *Q* is the octet length of *P*. The format of *B0* is illustrated in Figure 49.

Flags (1 octet)	Nonce (13 octets)	*Q* (2 octets)

Figure 49—Format of *B0*

The flags are formed by the following data:

— 1 reserved bit, which is set to '0';
— 1 bit flag for the associated data, which is zero;

— 3 bits to represent $(t - 2)/2$, which is 101 ($t = 12$);
— 3 bits to represent $q - 1$, which is 001 ($q = 2$).

The counter *Ctr(i)*, $i = 0, 1, ..., r$, is formed with 1-octet flags; 13-octet nonce *N*; and 2-octet integer *i*. The format of *Ctr(i)* is illustrated in Figure 50.

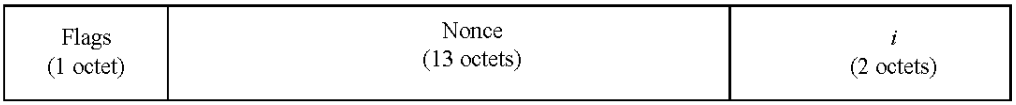


Figure 50—Format of counter *Ctr(i)*

The flags for *Ctr(i)* is 00000001.

The encapsulation of an MIS PDU consists of the following steps:

a) Fetch Transaction ID and FN from the MIS header.
b) Increment a positive number of *SN* to update the SN.
c) Construct the nonce, *N*, as described in 9.3.3.2.
d) Input *N* and *P* to AES-CCM generation-encryption process as specified in SP 800-38C. The *B0* and all the counter numbers are formed as described in Figure 49 and Figure 50, respectively.
e) Obtain the output, *C,* of AES-CCM.

9.3.3.3.2 Decapsulation

For a given SA, the prerequisites for AES-CCM decapsulation includes an encryption key MIEK, AES block cipher encryption block, and the parameters *t*, *n*, and *q*.

The decapsulation of a protected MIS PDU consists of the following steps:

a) Fetch Transaction ID and FN from the MIS header.
b) Fetch *SN* from the security TLV.
c) Construct the nonce, *N,* as described in 9.3.3.2.
d) Input *N* and *C* to AES-CCM decryption-verification process as specified in SP 800-38C. The *B0* and all the counter numbers are formed as described in 9.3.3.3.1.
e) Obtain the output, *P,* or "INVALID."

9.3.3.4 Format of security TLV

The ENCR_BLOCK data of the security TLV in a protected MIS message with AES-CCM is formed by SN and ciphertext C, which is the ciphertext of P and T (MIC). It is illustrated in Figure 51. Since MIC is carried in the ENCR_BLOCK data, the INTEGR_BLOCK in MIS_SPS_RECORD is not chosen for AES-CCM.

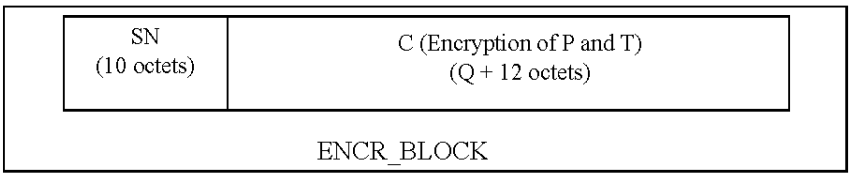


Figure 51—Security TLV for AES-CCM

9.3.4 MIS PDU protection by AES in CBC mode and HMAC-SHA-1-96

This ciphersuite includes two algorithms, encryption algorithm AES CBC and message authentication code algorithm HMAC-SHA1-96. When an MIS PDU is protected, encryption is applied first and an MIC is generated on the ciphertext. The MIC is 12 octets (96 bits). In order to use the ciphersuite, two keys (MIIK and MIEK) are used for encryption/decryption and generation/verification of MIC respectively.

9.3.4.1 Initialization vector for AES in CBC mode

Encryption using AES in CBC mode needs an initialization vector, IV, of 16 octets (128 bits), *IV* = (*IV0*, *IV1*, ..., *IV15*). It can be selected randomly when encryption is executed. It is also needed for decryption. Therefore, for each protected MIS PDU, an IV is included in ENCR_BLOCK as a part of security TLV.

9.3.4.2 Operational procedures in applying AES CBC and HMAC-SHA-1-96

9.3.4.2.1 Encapsulation

The encapsulation of an MIS PDU includes the following steps:

a) Select a 16-octet initialization vector, *IV*.
b) Pad the plaintext, *P,* to a length of a multiple of 16 octets (128 bits) so that the padded plaintext can be represented as in *n* blocks *P0*, *P1*, ..., *Pn-1*, each of which is 16 octets.
c) Apply AES CBC mode with *IV* and key, MIEK, on *P0*, *P1*, ..., *Pn-1* to obtain ciphertext *C0*, *C1*, ..., *Cn-1*.
d) Input *M* = *IV*||*C0*||*C1*||...||*Cn-1*, where '||' means concatenating, as the message and MIIK as the key to HMAC-SHA1. Here padding shall be needed when the input message length is not a multiple of 64 octets (512 bits). The most significant 12 octets of the output of HMAC-SHA1 is the MIC.
e) Output *C0*, *C1*, ..., *Cn-1,* and MIC.

9.3.4.2.2 Decapsulation

The decapsulation of a protected MIS PDU includes the following steps:

a) Fetch the ciphertext *C0*, *C1*, ..., *Cn-1* and the initialization vector, *IV*, from ENCR_BLOCK of security TLV.
b) Fetch the MIC from the INTG_BLOCK of security TLV.
c) Input *M* = *IV*||*C0*||*C1*||...||*Cn-1*, where '||" means concatenating, as the message and MIIK as the key to HMAC-SHA1. Here padding shall be needed when the input message length is not a multiple of 64 octets (512 bits). Compare the most significant 12 octets of the output of HMAC-SHA1 with the MIC. If they are identical, go to the next step. Otherwise, output "INVALID."
d) Input ciphertext, *C0*, *C1*, ..., *Cn-1*, and MIEK to AES CBC mode to obtain plaintext, *P0*, *P1*, ..., *Pn-1*.
e) Remove the padding if it is applied to obtain the plaintext, *P*.
f) Output *P*.

9.3.4.3 Format of security TLV

When an MIS PDU is protected by AES in CBC mode and HMAC-SHA1-96, both ENCR_BLOCK and INTG_BLOCK appear in the security TLV. The initialization vector IV and the ciphertext, *C0*, *C1*, ..., *Cn-1*, are included in ENCR_BLOCK and MIC is in INTG_BLOCK as shown in Figure 52.

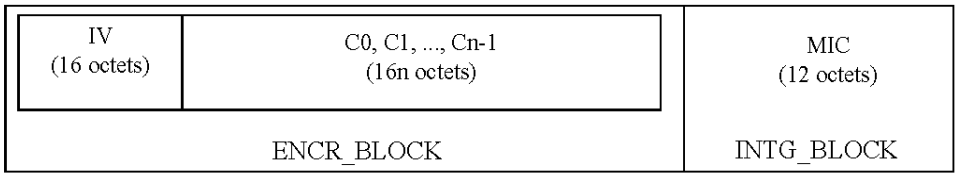


Figure 52—Security TLV for AES CBC and HMAC-SHA1-96

9.3.5 MIS PDU protection by HMAC-SHA-1-96

This ciphersuite includes one message authentication code algorithm, HMAC-SHA1-96. It generates a 12 octets (96 bits) MIC over the protected data using key MIIK.

9.3.5.1 MIC generation and verification

9.3.5.1.1 MIC generation

A MIC is generated in the following steps:

a) The data, *P*, to be protected is padded to a length of a multiple of 64 octets (512 bits).
b) Input the padded data and the key, MIIK, to HMAC-SHA1.
c) Obtain output of HMAC-SHA1.
d) Truncate the output of HMAC-SHA1 to obtain the most significant 96 bits as the MIC.
e) Output MIC.

9.3.5.1.2 MIC verification

A MIC is verified in the following steps:

a) Fetch the data, *P*, from the ENCR_BLOCK of security TLV. Pad it to a length of a multiple of 64 octets (512 bits).
b) Fetch the MIC from INTG_BLOCK of security TLV.
c) Input the padded data and the key, MIIK, to HMAC-SHA1.
d) Obtain output of HMAC-SHA1.
e) Compare the most significant 96 bits of the output of HMAC-SHA1 with MIC.
f) If they are identical, output is “VALID”; Otherwise, output is “INVALID.”

9.3.5.2 Format of security TLV

When an MIS PDU is protected by HMAC-SHA1-96, the plaintext is included in ENCR_BLOCK, even though it is not encrypted and the MIC is in the INTG_BLOCK as shown in Figure 53.

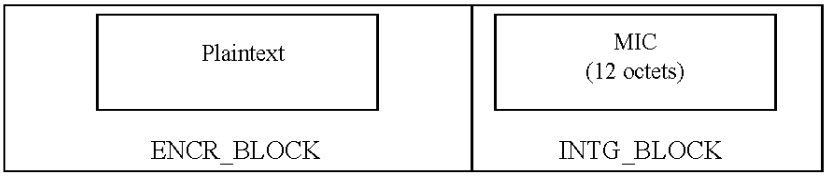


Figure 53—Security TLV for HMAC-SHA1-96

9.3.6 MIS PDU protection by AES-CMAC

This ciphersuite includes one MAC algorithm, AES-CMAC. It generates a 12 octets (96 bits) MIC over the protected data using key, MIIK.

9.3.6.1 MIC generation and verification

9.3.6.1.1 MIC generation

A MIC is generated in the following steps:

a) Input the data, *P*, to be protected and the key, MIIK, to AES-CMAC. (For AES-CMAC, the padding is specified as a part of the algorithm.)
b) Obtain output of AES-CMAC.
c) Truncate the 16 octets (128 bits) output of AES CMAC to obtain the most significant 96 bits as the MIC.
d) Output MIC.

9.3.6.1.2 MIC verification

A MIC is verified in the following steps:

a) Fetch the data, *P*, from the ENCR_BLOCK of security TLV.
b) Fetch the MIC from INTG_BLOCK of security TLV.
c) Input the data, *P*, to be protected and the key, MIIK, to AES-CMAC. (For AES-CMAC, the padding is specified as a part of the algorithm.)
d) Obtain output of AES-CMAC.
e) Compare the most significant 96 bits with the MIC.
f) If they are identical, output "VALID"; Otherwise, output "INVALID."

9.3.6.2 Format of security TLV

When an MIS PDU is protected by AES-CMAC, the plaintext is included in the ENCR_BLOCK, even though it is not encrypted and the MIC is in the INTG_BLOCK as shown in Figure 54.

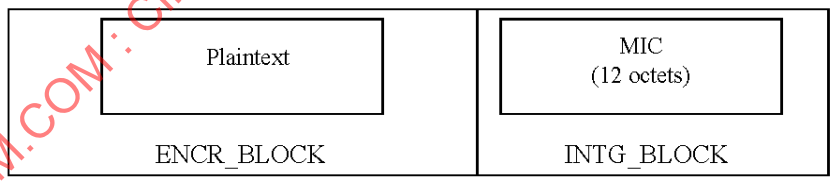


Figure 54—Security TLV for AES-CMAC

9.4 Common procedures

The following procedures are common for both (D)TLS- and EAP-generated MIS SAs.

9.4.1 Sending

For sending a remote MIS message in a protected manner, an MIS PDU is created in the following steps:

a) At the sender, which can be an MN or a PoS, the MIS user generates an MIS primitive and passes it to the MISF.

b) The MISF at the sender constructs an MIS PDU. If an MIS security association (SA) exists, then the MISF at the sender applies (D)TLS protection algorithms specified by the negotiated ciphersuite in the handshake to the MIS PDU and then encapsulates the protected MIS PDU in a security TLV. If no MIS SA exists, then the MIS PDU is passed to the transport protocol of the MIS message.

c) The security TLV is encapsulated in an MIS PDU with the S bit in the MIS header set to '1'.

d) The MIS PDU is passed to the transport protocol of the MIS message.

9.4.2 Receiving

For receiving a protected MIS message from a remote entity, the protected MIS PDU is processed in the following steps:

a) At the receiver, which can be an MN or a PoS, the MISF receives a protected MIS PDU from the transport protocol of the MIS message.

b) If the S bit is set to '1' in the header, the MISF processes security TLV and extracts the plaintext MIS PDU. Otherwise, it takes MIS PDU as is.

c) The MISF creates an MIS primitive from MIS PDU and passes it to the MIS user at the receiver.

The processing steps at the sender and receiver are described in Figure 55.

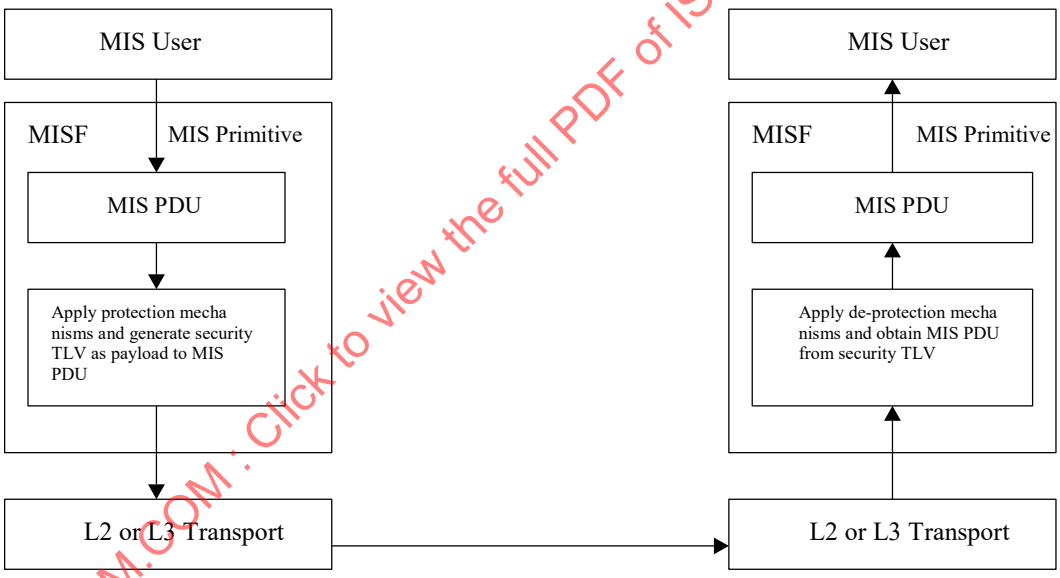


Figure 55—Sending and receiving protected MIS PDU

The transport protocol entities to be associated with an MIS SA are MISF peers and are identified by MISF identifiers. Therefore, the transport address of an MISF can change over the lifetime of an MIS SA as long as the mapping between the transport address and the MISF identifier of an MISF is maintained.

9.5 Group manipulation for group addressed messages

A group addressed message is an MIS message sent to a group of recipients. A recipient of a group addressed message can be a mobile node (MN) or a point of service (PoS). Each group is identified by an MISF Group ID. A group is dynamic when some of the group members leave and the new member joins. The group is managed through group manipulation commands.

A series of group addressed messages may follow a group manipulation command that defines a target group of recipients. A group addressed message is sent, for instance, by a PoS to instruct the group that the members should handover to a PoA or that they should update their configuration parameters. A payload of a group addressed message can be protected (encrypted) using the SA derived from the MGK. The following describes group manipulation commands and group addressed messages:

— A group manipulation command is issued by a group manager to instruct recipients to join or leave a group. A group manipulation command should also be used to update a group key stored at a recipient. Group manipulation commands are carried in an MIS_Push_Group_Manipulate request or indication message or an MIS_Pull_Group_Manipulate response message. An MIS_Push_Group_Manipulate request or indication message is either unicast or multicast. An MIS_Pull_Group_Manipulate response message is unicast. Those messages are digitally signed by the originating MISF. A group is identified by an MISF Group ID and associated with a multicast transport. The address used by this multicast transport can be provided by the group manipulation command itself. Note that a recipient over a multicast transport group membership may not be in the group addressed by a group manipulation command.

— A group addressed message is sent to instruct the recipients to take an action. The MISF Group ID of the target group is set to the Destination MISF ID field in the group addressed message. A group addressed message is sent using a multicast transport associated with the MISF Group ID. A group addressed message contains two types of payload: protected or unprotected. If a payload is protected, it uses AES-CCM mode with a key derived from the current group key. Whether it is protected or not, the payload of a group addressed message sent by a PoS shall be authenticated by a digital signature of the PoS, and the payload of a group addressed message sent by an MN may be authenticated by a digital signature of the MN.

Note that a recipient over the multicast transport may not be in the target group as specified by the group manipulation command.

9.5.1 Key distribution for multicast MIS message protection

Group addressed messages are protected by media independent group session key (MIGSK). An MGK is distributed through a group manipulation command. The command carries a special data field, called group key block (GKB) that includes the encrypted MGK.

It is assumed that each potential recipient is provisioned with a set of device keys. Provisioning of the device keys is done through out-of-band mechanisms. GM shall have the information about the device keys in order to make sure that intended device keys are revoked. Depending on the members in the group, an MGK is encrypted by a specific set of device keys so that each member in the group can decrypt one of the encrypted versions to obtain the MGK. A recipient belonging to the group can use one of its device keys to recover MGK, from which it can derive the MIGSK used to protect the group addressed messages.

9.5.1.1 Device key assignment through a group management tree

The device key assignment mechanism specified in this standard is based on a binary key structure, called a group management tree (a precedent of this mechanism can be found in IETF RFC 2627). A group management tree is a binary tree of depth d, where d is a system constant. A group manager, at its initialization period, shall generate a group management tree whose depth is less than 256. The root of the group management tree is called a level 0 node. A node is a logical entity in a binary tree for which each leaf node represents a potential recipient of group addressed messages sent to a group created by the group manager of the group management tree. Each recipient is a device, which can be an MN or a PoS. At level k, there are 2^k nodes, $0 \leq k \leq d$. Each level k node can be represented as a k-bit string. The string is called the index of the node. For example, when $d > 1$, the level 2 nodes are represented by the indices 00, 01, 10, 11. Each node is assigned to a key, called a node key, and is identified by the Node Index. For example, the node keys assigned to level 2 nodes are denoted as k<00>, k<01>, k<10>, k<11>.

For a group management tree of depth d, all the level d nodes are called leaf nodes. Each potential recipient (an MN or a PoS) is assigned to a leaf node. A recipient assigned to a leaf node with index $x_0x_1…x_{d-1}$, is provisioned with device keys k<x_0>, k<x_0x_1>, …, k<$x_0x_1…x_{d-1}$>. Figure 56 illustrates a group management tree of depth 3. Table 26 lists the device key assignment for each recipient represented by the leaf Node Index.

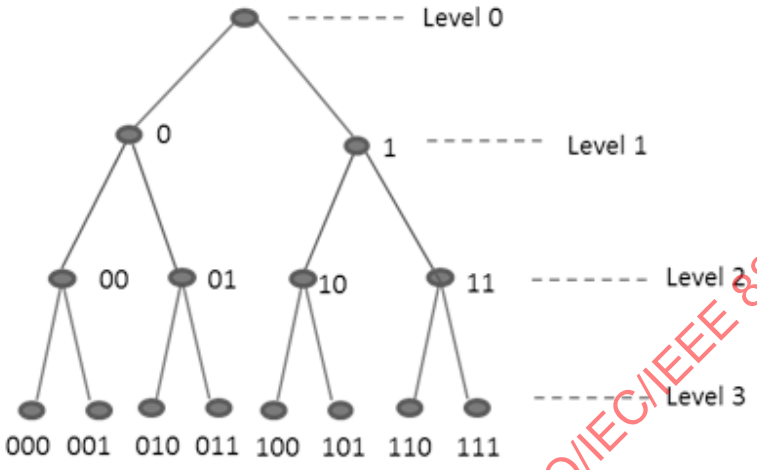


Figure 56—A group of management tree of depth 3

Table 26—Device key assignments for recipients through a depth-3 group management tree

Node	Device keys
000	k<>, k<0>, k<00>, k<000>
001	k<>, k<0>, k<00>, k<001>
010	k<>, k<0>, k<01>, k<010>
011	k<>, k<0>, k<01>, k<011>
100	k<>, k<1>, k<10>, k<100>
101	k<>, k<1>, k<10>, k<101>
110	k<>, k<1>, k<11>, k<110>
111	k<>, k<1>, k<11>, k<111>

When determining the system constant d for a group management tree, 2^d shall be no less than the number of all the recipients to be grouped in the system. In this standard, each leaf index is represented as an octet string. Therefore, the depth of a group management tree is chosen as a multiple of 8.

For each leaf node, the Node Index is considered as the binary representative of an integer, called leaf number. For example, in the group management tree in Figure 56, the leaf nodes with the Node Index 000 has leaf number 0, 001 has leaf number 1, 010 has leaf number 2, etc.

9.5.1.2 Complete subtree

A complete (depth-k) subtree is a subtree with 2^k leaf nodes such that their indices have common prefix of d-k bits. For the group management tree in Figure 57, nodes represented with indices 000 and 001 form a depth-1 complete subtree, while nodes represented with indices 000, 001, 010, and 011 form a depth-2 complete subtree.

When the device keys are assigned through a group management tree as introduced in 9.5.1.1, all the recipients corresponding to the leaf nodes in a complete subtree share a common node key (as one of the device keys), determined by the d-k bit prefix. In the example above, the recipients corresponding to the leaf nodes represented with indices 000 and 001 share node key k<00>. Therefore, a subset of members in a group corresponds to the leaf nodes in a complete subtree that can decrypt the MGK encrypted by the common shared key.

In order to distribute an MGK to a group of recipients, the first step is to sort the corresponding leaf nodes to non-overlap complete subtrees so that each leaf node belongs to one complete subtree. Note that a single leaf node can be considered as a depth-0 complete subtree. In this case, the MGK is encrypted by the minimum number of keys for the group.

In the example illustrated in Figure 57, the group represented by the leaf nodes 000, 001, 010, 011, 101, and 111 can be sorted as one depth-2 complete subtree containing leaf nodes 000, 001, 010, 011, and two depth-0 complete subtrees containing leaf nodes 101 and 111, respectively. A complete subtree can be identified by its root node. The aforementioned complete subtrees are identified by node 0, 101, and 111, respectively. For the recipients in such a group, the MGK shall be encrypted by k<0>, k<101>, and k<111>.

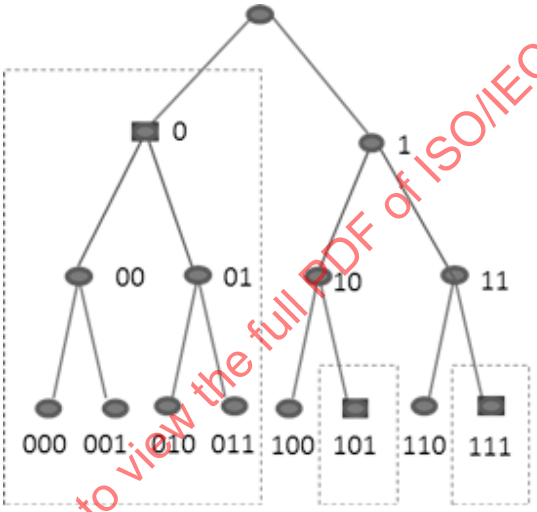


Figure 57—Three complete subtrees for the group with nodes 000, 001, 010, 011, 101, and 111

The algorithm of sorting a group of leaf nodes to non-overlapping complete subtree is introduced in the following:

```
def CreateCompleteSubtree(I, T, R):
        # Input I: List of indices of leaf nodes to be included in the group
        # Input T: The entire tree that covers all leaf nodes
        # Input R: Root node of the entire tree
        # Output S: Complete Subtree for the group.
        S=[]
        def check(n):
                # Input n: subtree root node
                # Output 0, 1
```

```
            #  0 : Some node in the subtree is a non-member of the group.
            #  1 : All nodes in the subtree are members of the group.
            if n.left==None and n.right==None:  # n is leaf
                    if n.index.val in I:
                            S.append(n)
                            return 1
                    return 0
            # n is non-leaf
            lval=check(n.left)
            rval=check(n.right)
            if lval*rval>0:
                    S.remove(n.left)
                    S.remove(n.right)
                    S.append(n)
                    return 1
            return 0
    check(R)
    return S
```

9.5.2 GKB generation by the complete subtree method

A GKB contains a complete subtree (CompleteSubtree) part and a group key data (GroupKeyData) part. A GroupKeyData part appears when a GKB is used to deliver a group key. When GroupKeyData part does not exist, CompleteSubtree part is used to identify the groups.

A complete subtree part identifies all the complete subtrees covered by the group such that each member represented by a leaf node in the group belongs to one and only one complete subtree. Each complete subtree is identified by its root node. The index of the root node is represented as a Node Index. A Node Index is a pair of binary strings: Node Depth and Node Index Value. The first parameter, Node Depth, indicates the binary length of the index of the root node represented by one octet. The second parameter, Node Index Value, is a binary string representing the index of the root node.

The Node Depth is represented by an octet. The corresponding integer L is the binary length of the index of the root node representing the complete subtree. The Node Index Value is represented by *ceil(L/8)* octets, where *ceil(x)* is the ceiling function, which takes the minimum integer, which is larger than or equal to x. For example, *ceil(1.2) = 2*. With the ceiling function, to demonstrate, if the value of a Node Depth L is between 1 and 8, the size of the binary string used to represent the Node Index Value is 1 octet (= 8 bits). If L is between 9 and 16, the size of the binary string used to represent the Node Index Value is 2 octets (= 16 bits). A Node Index Value is left aligned in the network byte order. A Node Index Value shall have a zero padding added to the right. An example of Node Index is (0x05, 0b10011000). This Node Index represents the node with index '10011,' which is the root node of the complete subtree in a group management tree. The depth of the group management tree is a system parameter and configured to the applications. If the group management tree has depth 8, then the root node with index '10011' identifies a depth-3 complete subtree. Another example of Node Index is (0x0e, 0b1100101000011100), which represents the node with index '11001010000111.' If it is in a depth-16 group management tree, then the node identifies a depth-2 complete subtree.

A complete subtree part is a list of Node Indices. The Node Indices are ordered based on the following rule. Let (L_1, I_1) and (L_2, I_2) be two Node Indices. (L_1, I_1) appears in the front of (L_2, I_2), if and only if $Int(d, I_1) < Int(d, I_2)$, where $Int(d, I_h)$ is a function to convert binary index I_h to a d-bit long integer. It uses I_h as the

binary values in the most significant bits of a d-bit integer. For example, if d = 8, (0x05, 0b10011000) is converted to $2^7 + 2^4 + 2^3 = 152$. When another Node Index is (0x08, 0b00011001) (with index 00011001 and convert to $2^4 + 2^3 + 1 = 25$), then (0x05, 0b10011000) appears first. A group key data part of a GKB is a sequence of ciphertexts of the encrypted group key, where a group key is encrypted by node keys corresponding to all the root nodes of the complete subtrees covered by the group. There is a one-one correspondence between the complete subtree part and the group key data part in a GKB. The number of Node Indices, i.e., the number of complete subtrees, in the complete subtree part is equal to the number of ciphertexts of the encrypted group key in the group key data part. The n-th ciphertext of the group key is encrypted by the node key corresponding to the n-th Node Index. AES_Key_Wrapping-128 or AES_ECB-128 is used for the encryption algorithm.

In the example illustrated in Figure 57, the group with nodes 000, 001, 010, 011, 101, and 111 is covered by three complete subtrees at root node 0, 101, and 111. Therefore, the complete subtree part in the GKB includes Node Indices for node 0, 101, and 111. The key data part includes the ciphertexts of MGK encrypted by k(0), k(101), and k(111) as indicated in Figure 58.

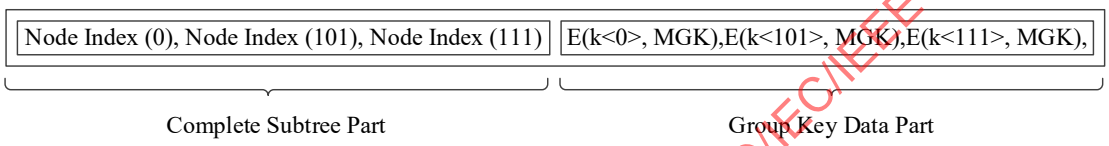


Figure 58—GKB for the group with nodes 000, 001, 010, 011, 101, and 111

Notice that the example is for illustration purpose. The group management tree has depth 3, not a multiple of 8. The Node Index Value is not illustrated with the binary string in Figure 57. In fact, Node Index (0) can be represented as (0x01, 0b000000000), Node Index (101) as (0x03, 0b10100000), and Node Index (111) as (0x03, 0b11100000).

There are two ways group members can be identified in a GKB when only CompleteSubtree exists in a GKB. The following methods are used to identify the group members:

Method 1: The set of leaf nodes specified by the complete subtree part of the GKB represents the members who belong to the group.

Method 2: The set of leaf nodes specified in the complete subtree part of the GKB represents the members who do not belong to the group. In other words, the complete subtree part represents the complement set of the leaf nodes.

For example, in a depth-3 group management tree, the set of all the leaf nodes is S = {000, 001, 010, 011, 100, 101, 110, 111} and the group consists of members with leaf nodes in a set is; A = {000, 001, 010, 011, 100}. When Method 1 is used, the complete subtree part shall represent set A, while when Method 2 is used, the complete subtree part shall represent S − A = {101, 110, 111}.

In order for a recipient to distinguish the two methods, a group manipulation command accompanies a flag named ComplementSubtreeFlag. If the flag is 0, Method 1 is used. If the flag is 1, Method 2 is used. The ComplementSubtreeFlag thus helps the recipient to correctly interpret the complete subtree part of a GKB.

A group manager has a component called GKB Generator. A GKB Generator receives all the device keys assigned to all the recipients associated to a group and an MGK. The MGK is a master group key for that group. Recipients and the group manager can generate a media independent group session key (MIGSK) from MGK (see 9.6). The mechanism to provide all device keys to the GKB Generator is out of the scope of this specification. This mechanism can just encompass the explicit provision of the device keys to the

GKB Generator or the random seed used to derive them. On receiving those data, a GKB Generator outputs a GKB, or several GKBs.

9.5.2.1 Master group key wrapping

For a specific group, to encapsulate an MGK in GKB, an MISF of a PoS first identifies the complete subtrees for the group using the algorithm described in 9.5.1.2. Let N be the number of complete subtrees contained in the CompleteSubtree. The MISF encrypts the MGK using each of the N node keys corresponding to the complete subtrees and obtains an ordered list of N encrypted versions of the MGK such that i-th element of the ordered list corresponds to the ciphertext encrypted by the node key of the i-th complete subtree in the CompleteSubtree for all integer i in $1 \leq i \leq N$. The obtained ordered list becomes the GroupKeyData. Figure 60 presents a flow diagram of the master group key wrapping procedure, known as MasterGroupKeyWrapping in the rest of this document.

The MISF selects a string, and derives media independent group key verification key (MIGKVK) from MGK as described in 9.6.1.

The MISF derives MAC of the string using MIGKVK. The string and MAC become VerifyGroupCode for MGK. Figure 59 presents a flow diagram of the verify group code generation procedure.

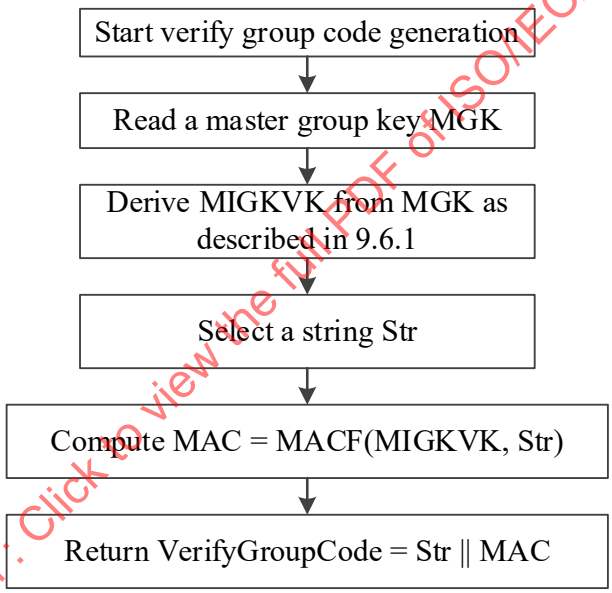


Figure 59—Flow diagram of the verify group code generation

For example, for depth-3 group management tree introduced in 9.5.1.2, let CompleteSubtree be given as ((0x01, 0b00000000), (0x03, 0b10100000), (0x03, 0b11100000)). Let X be an MGK. Then, the GroupKeyData in the GKB for the given CompleteSubtree is (Enc(k<0>, X), Enc(k<101>, X), Enc(k<111>, X)) where Enc is an encapsulation algorithm.

If a master group key is not distributed in the GKB, the master group key wrapping step and the verify group code generating step should be omitted. If an authenticated encryption (e.g., AES-Key_Wrapping) is used to generate GroupKeyData, the verify group code generating step shall be omitted.

After generating the GroupKeyData, The MISF has a choice of replacing the list of Node Indices in the Complete Subtree by the Bloom Filter of the list. If the MISF uses the Bloom Filter as the Complete Subtree or omits the Complete Subtree, the GroupKeyData shall be generated by the authenticated encryption or the VerifyGroupCode shall be attached.

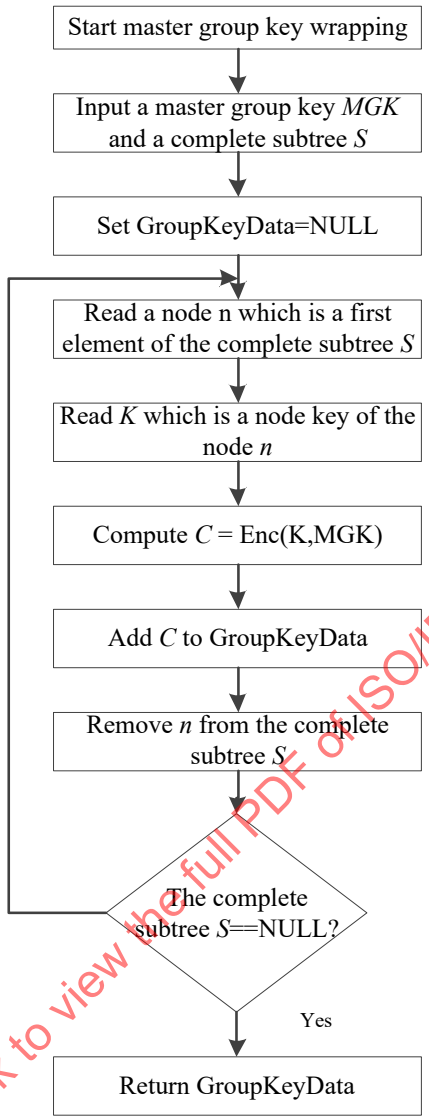


Figure 60—Flow diagram of the group key wrapping

9.5.2.2 Master group key unwrapping

The master group key unwrapping procedure for a GKB is described as follows and it is shown in Figure 61:

— An MISF in the recipient of GKB checks GroupKeyData. If GroupKeyData does not exist, then start the *no group key procedure,* else continue with the next step.

— If Group Key Data is contained, the MISF in the recipient of GKB checks CompleteSubtree. If CompleteSubtree contains a list of Node Index (see Figure 62), go to *master group key unwrapping procedure 1* in 9.5.2.2.1. Else if CompleteSubtree contains a Bloom Filter, go to *master group key unwrapping procedure 2* in 9.5.2.2.2. Else if CompleteSubtree is not given, go to *master group key unwrapping procedure 3* in 9.5.2.2.3.

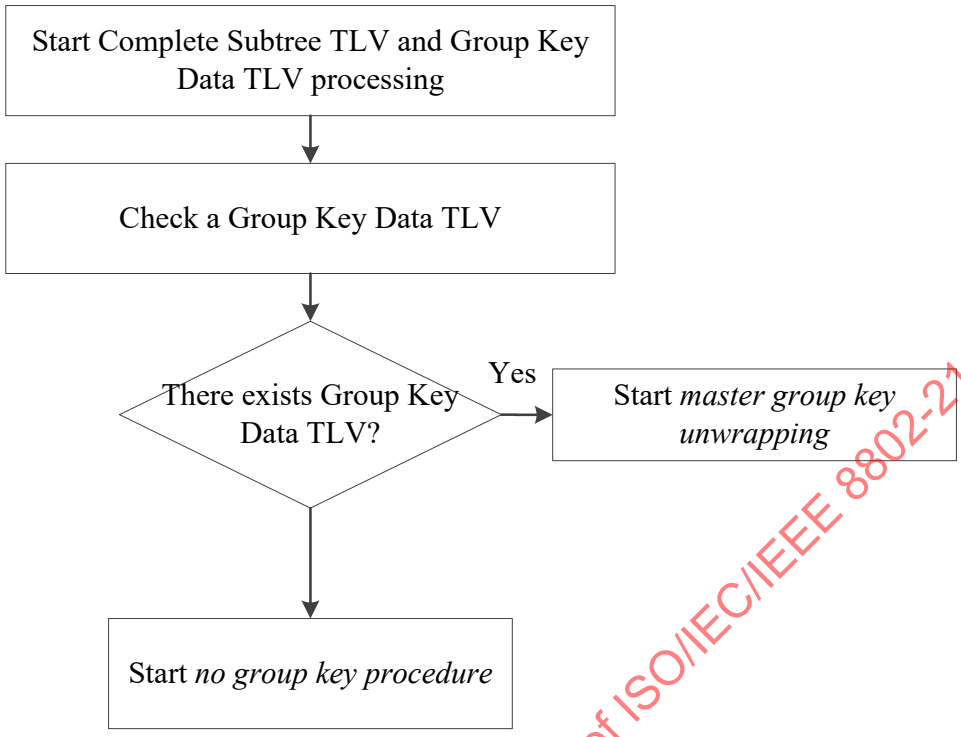


Figure 61—Selection of *master group key unwrapping* or *no group key procedures*

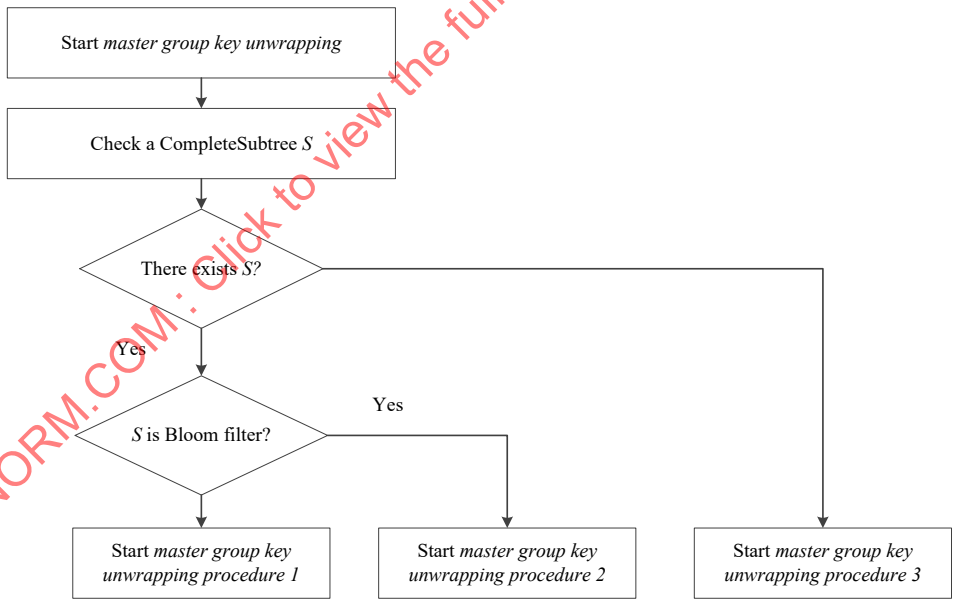


Figure 62—Flow diagram of the group key unwrapping

9.5.2.2.1 No group key data procedure

In a scenario when a CompleteSubtree is provided as a list of Node Index but a GroupKeyData is not available, *no group key procedure* shall be performed as follows:

a) The procedure reads a CompleteSubtree *S*.
b) The procedure reads a list of Node Indices *I* for device keys of the recipient.
c) The procedure finds *n* in *I* such that *n* is *i-th* element of the complete subtree *S*. If it succeeds to find such *n*, it returns Success. Else it returns Fail.

Figure 63 is a flow diagram of the *no group key data procedure*.

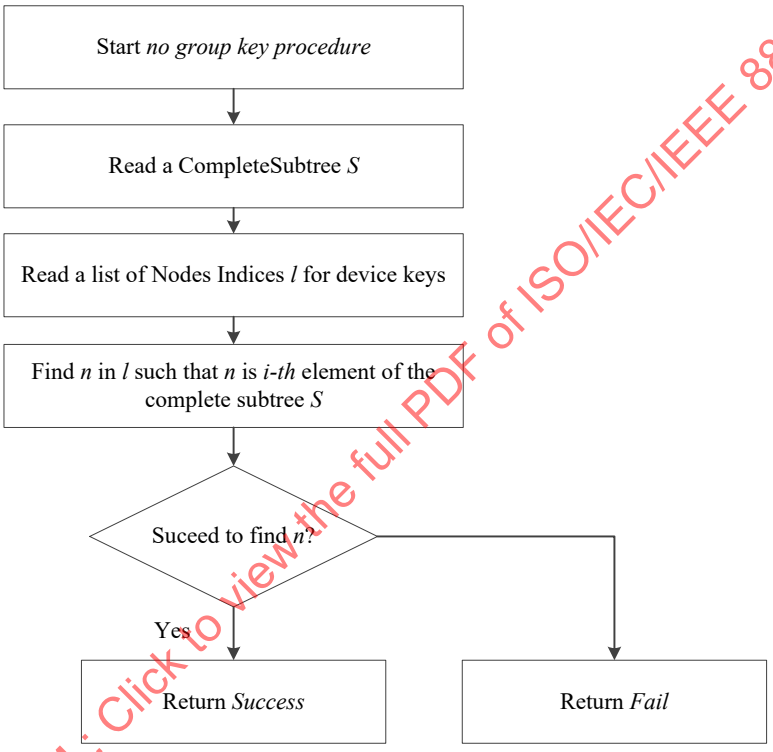


Figure 63—Flow diagram of *no group key data procedure*

9.5.2.2.2 Master group key unwrapping procedure 1

In a scenario when a CompleteSubtree is provided as a list of Node Index and a GroupKeyData is available, *master group key unwrapping procedure 1* shall be performed as follows:

a) The procedure reads a CompleteSubtree S and a GroupKeyData KB.
b) The procedure reads a list of Node Indices *I* for device keys of the recipient.
c) The procedure finds *n* in *I* such that *n* is *i-th* element of the complete subtree *S*. If it succeeds to find such *n*, then it extracts a device key *k* that corresponds to *n*. Else it returns Fail.
d) The procedure reads an encrypted group key *C* from *i-th* element of *KB*.
e) The procedure decrypts *C* using *k*. If the decryption is successful, it returns Success and the result of the decryption as a master group key, *MGK*. Else it returns Fail.

Figure 64 is a flow diagram of the *master group key unwrapping procedure 1*.

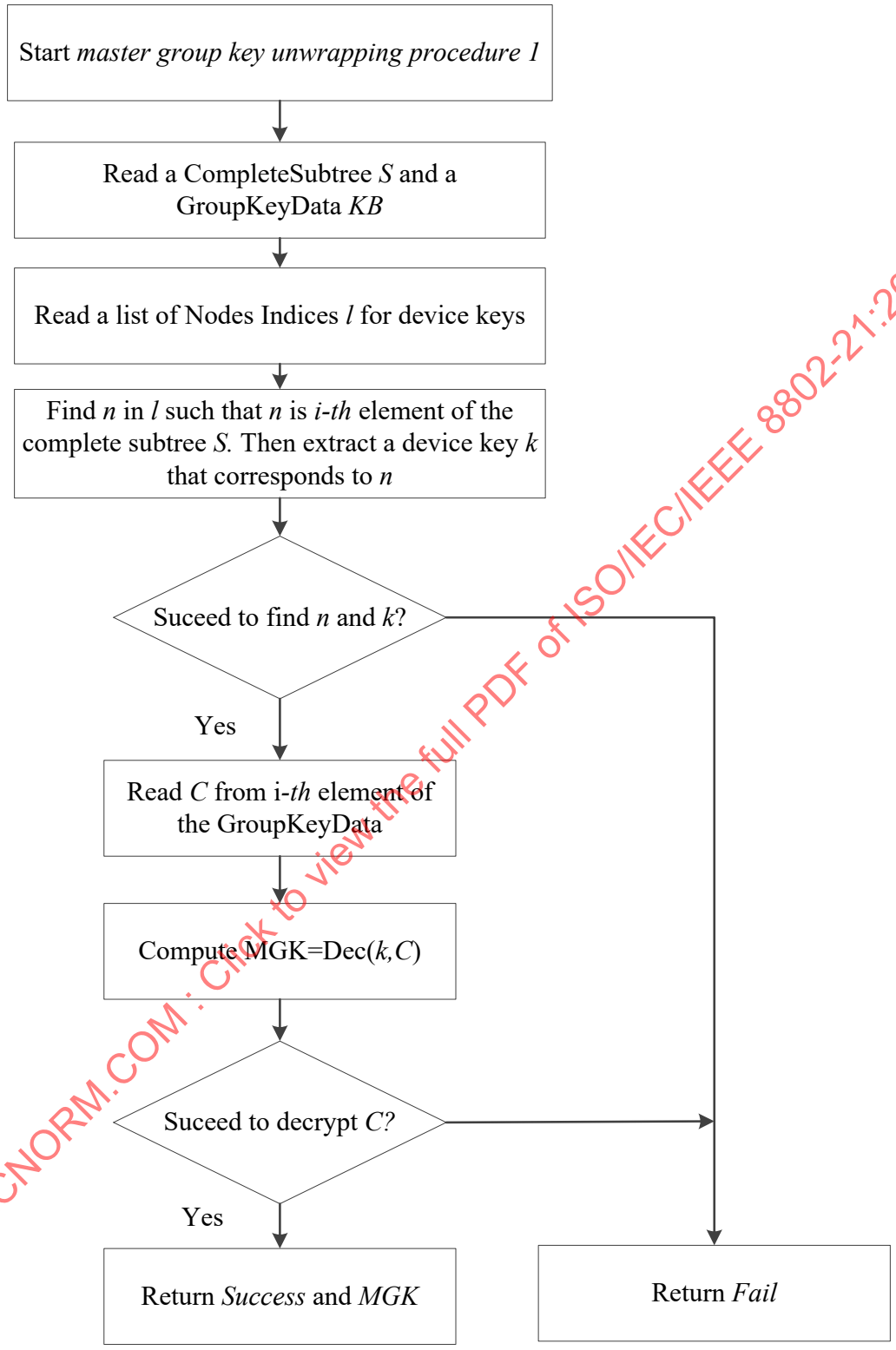


Figure 64—Flow diagram of the *master group key unwrapping procedure 1*

9.5.2.2.3 Master group key unwrapping procedure 2

In a scenario when a CompleteSubtree is provided as a Bloom Filter and a GroupKeyData is available, *master group key unwrapping procedure 2* shall be performed as follows (shown in Figure 65):

a) The procedure reads a Bloom Filter *BF* in a CompleteSubtree, a GroupKeyData *KB*, and VerifyGroupCode *VGC* (optional).
b) The procedure reads a list of Node Indices *I* for device keys of the recipient, and sets $i = 1$.
c) The procedure sets *n* to *i*-th element of *I*.
d) The procedure checks whether *n* matches *BF*. If it succeeds to the matching, then it extracts a device key *k* that corresponds to *n,* and it sets $j = 1$. If it fails to the matching, go to Step g).
e) The procedure sets *C* to *j*-th element of *KB*.
f) The procedure decrypts *C* using *k*. The MISF checks the result of decryption using VerifyGroupCode (optional). If the decryption is successful, it returns *Success* and the result of the decryption as a master group key, *MGK*. Else if *C* is not the last element of *KB*, it sets $j = j + 1$, and goes to Step e).
g) If *n* is not the last element of *I*, the procedure sets $i = i + 1$ and goes to Step c). Else it returns Fail.

9.5.2.2.4 Master group key unwrapping procedure 3

In scenario when CompleteSubtree is not provided and a GroupKeyData is available, *master group key unwrapping procedure 3* shall be performed as follows (shown in Figure 66):

a) The procedure reads a GroupKeyData *KB*, and VerifyGroupCode *VGC* (optional).
b) The procedure reads a list of device keys *DK* of the recipient, and sets $i = 1$.
c) The procedure sets *k* to *i*-th element of *DK*, and sets $j = 1$.
d) The procedure sets *C* to *j*-th element of *KB*.
e) The procedure decrypts *C* using *k*. The MISF checks the result of decryption using VerifyGroupCode (optional). If the decryption is successful, it returns *Success* and the result of the decryption as a master group key, *MGK*. Else if *C* is not the last element of *KB*, it sets $j = j + 1$, and goes to Step d).
f) If *n* is not the last element of *I*, the procedure sets $i = i + 1$ and goes to Step c). Else it returns *Fail*.

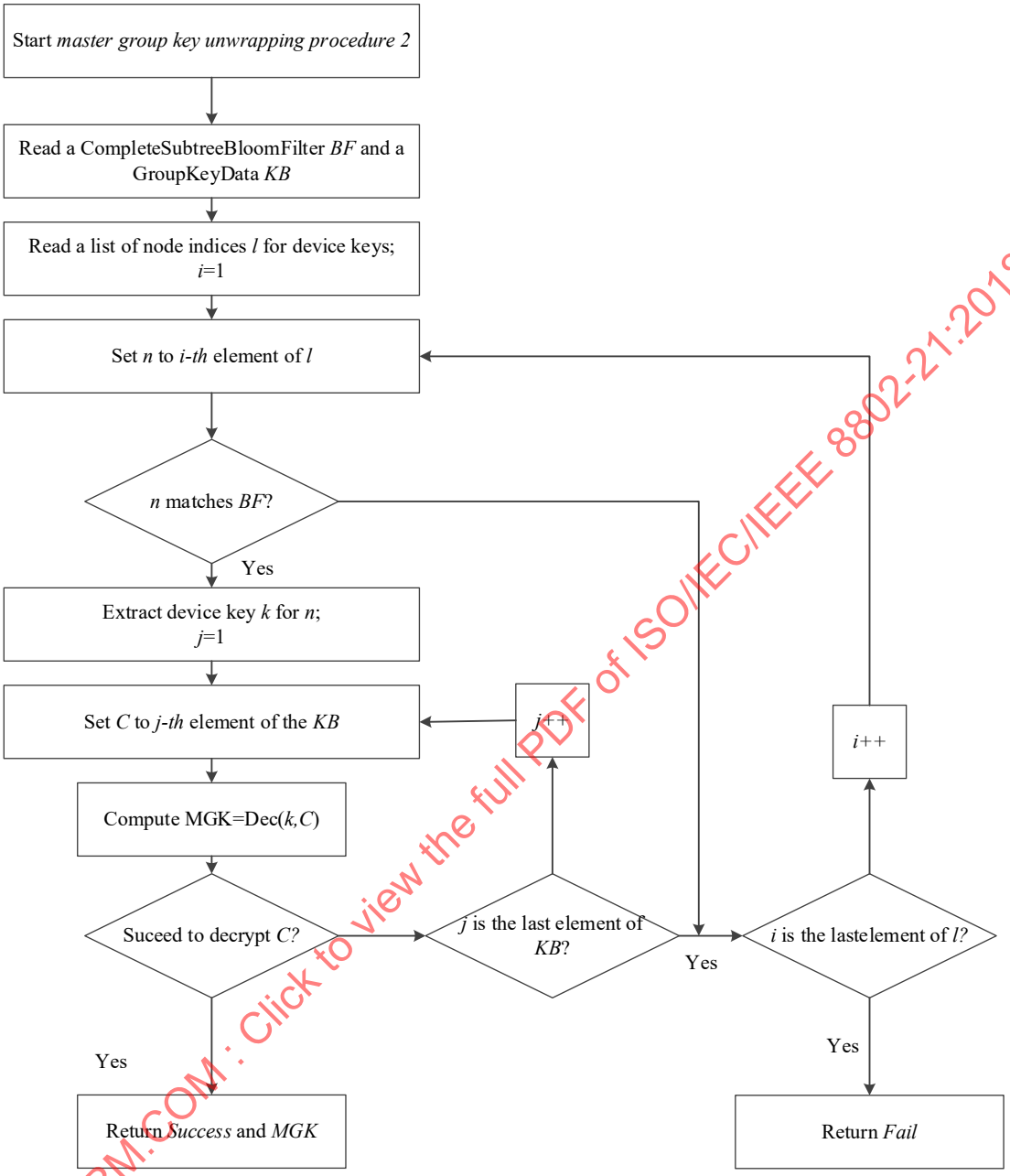


Figure 65—Flow diagram of the *master group key unwrapping procedure 2*

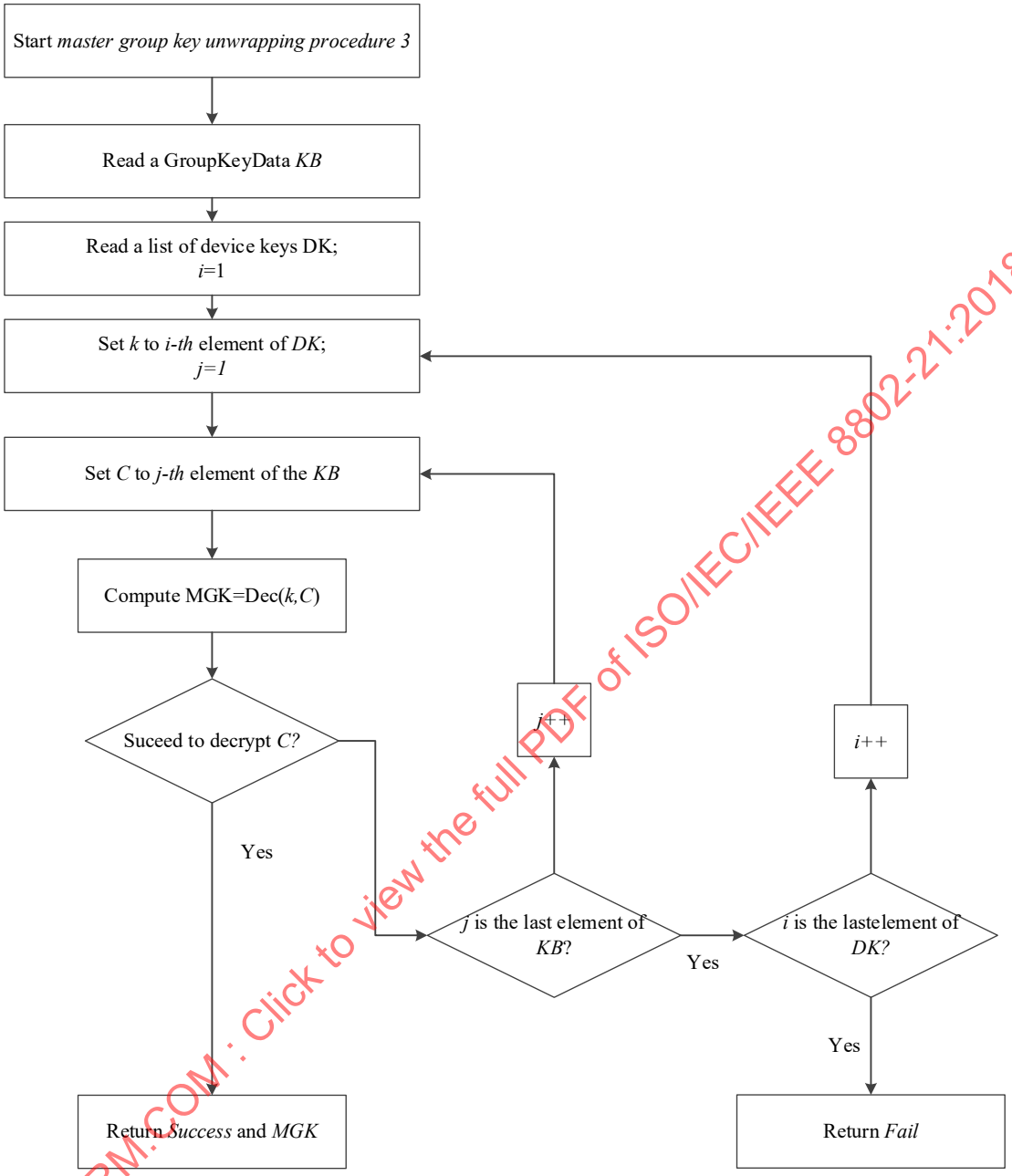


Figure 66—Flow diagram of the *master group key unwrapping procedure 3*

9.5.2.3 Fragmented GKB

Assume that each GKB has a length limit. When the length of a GKB exceeded the limit, fragmentation is needed. That is, instead of one GKB, multiple GKB fragments are transmitted. Notice that the number of ciphertexts in a GKB, which is the major factor in determining the length of a GKB, is determined by the number of complete subtrees the group covers. If the limit is represented by the number of complete subtrees and denoted as *GkbFragmentThreshold*, when a GKB contains more than *GkbFragmentThreshold* of Complete Subtrees, it is fragmented into multiple GKB fragments such that each GKB fragment contains at most *GkbFragmentThreshold* Complete Subtrees. When a GKB is transmitted in multiple fragments, each fragment is identified by a Subgroup Range. A recipient processes one fragment with Subgroup Range

that covers its leaf number. A Subgroup Range is defined by a pair of integers [a, b] such that a potential recipient with leaf number r shall look into the fragment if a ≤ r ≤ b. Each GKB fragment is associated with exactly one Subgroup Range. For example, for a depth-3 group management tree as illustrated in Figure 57, the Subgroup Range [0, 3] corresponds to the leaf nodes with indices 000, 001, 010, 011. The Subgroup Ranges for GKB fragments of the same GKB satisfy all of the following conditions:

a) Union of all Subgroup Ranges is equal to the group range defined as the range of the leaf indices of the group management tree. For example, for a depth-3 group management tree as illustrated in Figure 57, the Subgroup Range can be [0, 3], [4, 5], [6, 7].
b) Intersection of any two Subgroup Ranges is empty.
c) Each complete subtree shall be in one and only one Subgroup Range.

GKB fragment length should not exceed the MIS fragment length.

The default algorithm by which Complete Subtrees and Subgroup Ranges satisfy these conditions is defined as follows.

```
def CreateCompleteSubtreeFragments(I, T, R, M):
      # Input I: List of indices of leaf nodes to be included in the group
      # Input T: The entire tree that covers all leaf nodes
      # Input R: Root node of the entire tree
      # Input M: Maximum number of subtrees in per fragment
      # Output O: List of (S, minr, maxr):
      #   S: Subtrees covering the group.
      #   minr: Lower bound of Subgroup Range
      #   maxr: Upper bound of Subgroup Range
      O=[]
      S=[]
      depth=int(math.log(len(T)+1,2)-1)

      def rightmost_leaf_number(n):
              # Input n: subtree root node
              # Output y: rightmost leaf number under the subtree
              h=n.index.len  # hierarchy level of node n
              x=int(n.index.val, 2)  # node index in decimal
              y=(x+1)*(2**(depth-h))-1
              return y

      def check(n):
              # Input n: subtree root node
              # Output 0, 1
              #  0 : Some node in the subtree is a non-member of the group.
              #  1 : All nodes in the subtree are members of the group.
              global minr
              rv=0
              if n.left==None and n.right==None:  # n is leaf
```

```
            if n.index.val in I:
                S.append(n)
                 return 1
        return 0
        # n is non-leaf
        lval=check(n.left)
        rval=check(n.right)
        if lval*rval>0:
            S.remove(n.left)
            S.remove(n.right)
            S.append(n)
            rv=1
        elif lval+rval>0:
            if len(S) > M: # one fragment is ready
                maxr=rightmost_leaf_number(S[M-1])
                O.append((S[0:M], minr, maxr))
                S[0:M]=[]   # Remove the appended subtrees
                minr=maxr+1
                rv=0
            if n==R: # Root node. len(S)>0
                maxr=2**depth-1
                O.append((S, minr, maxr))
            return rv
check(R)
return(0)
```

9.5.3 Secure group manipulation procedures

Figure 67 illustrates group manipulation command distribution initiated by a PoS with group manager via a multicast transport. The MIS user of the PoS with group manager generates an MIS_Push_Group_Manipulate.request, described in 7.4.21, and then it passes the request to the MISF of the group manager. Upon receiving the request, the MISF generates MIS_Push_Group_Manipulate indication (note that the decision on sending an indication message or a request message depends on the ResponseFlag parameter of the MIS_Push_Group_Manipulate.request primitive), described in 8.6.1.25, and sends it to the recipients via multicast mechanisms. When a recipient receives the MIS_Push_Group_Manipulate indication message, the MISF of the recipient processes the message. After processing the message, the MISF sends MIS_Push_Group_Manipulate.indication to the MIS user of the recipient.

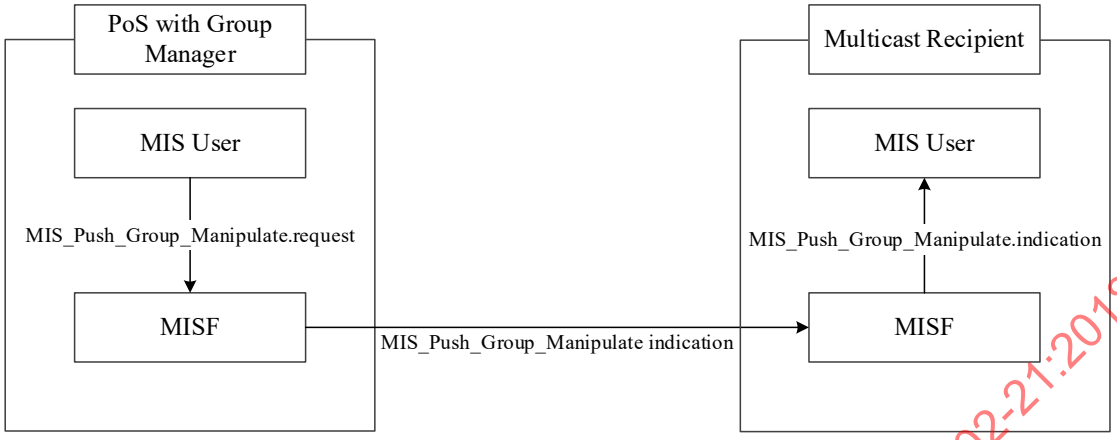


Figure 67—Example of group manipulation distribution using multicast mechanisms

9.5.3.1 Sending procedures for group manipulation commands

9.5.3.1.1 Group manager

A group manager is an MIS user of a PoS. In the PoS, the MISF needs the MGK to encrypt service specific TLVs in group manipulation and group addressed commands. MISF obtains the MGK via MIS_Push_Group_Manipulate.request primitive. Required components in a group manager relevant to group manipulation and group addressed commands are listed as follows:

— A GKB Generator. This component is composed of CreateCompleteSubtree procedure (see 9.5.1.2) or CreateCompleteSubtreeFragments procedure (see 9.5.2.3), and MasterGroupKeyWrapping procedure (see 9.5.2.1). If GKB is always not fragmented, CreateCompleteSubtree procedure should be used. Otherwise, CreateCompleteSubtreeFragments procedure should be used.

— A *Tree Information Base* (of type GRP_MGT_TREE_INFO_BASE as defined in Table E.25). This information base contains all the pairs of an MISF ID and a corresponding leaf number, and all the pairs of a Node Index and a corresponding node key.

— A *Group Information Base* (of type MANAGED_GROUP_INFO_BASE as defined in Table E.25). This information base stores the information about groups which are managed by the PoS with group manager. It stores tuples of an MISF Group ID, the MISF IDs of the group members and, optionally, the MGK and the multicast transport addresses assigned to a group.

A Flow diagram of the generation process of the GKB parameters is given in Figure 68. The group manager (MIS user) generates MIS_Push_Group_Manipulate.request described in 7.4.21.1 as follows:

a) Choose an MISF Group ID and group members to manipulate. If the group does not exist already, choose an MISF Group ID by consulting with the *Group Information Base*.

b) If necessary, update the membership information, the MGK, and the transport address in the *Group Information Base.*

c) Define TargetIdentifier:

— Set the MISF Group ID chosen in step a) as TargetIdentifier.

d) Define CompleteSubtree and SubgroupRange:

1) Determine ComplementSubtreeFlag value.

2) If ComplementSubtreeFlag = 0 or CompleteSubtree is not present.

i) If CreateCompleteSubtree procedure is used, the MIS user sends leaf numbers that correspond with MISF IDs of the group members and all Node Indices which represent

the group management tree to the CreateCompleteSubtree procedure and receives CompleteSubtree for the GKB.

ii) If CreateCompleteSubtreeFragments procedure is used, the MIS user sends leaf numbers that correspond with MISF IDs of the group members, all Node Indices which represent the group management tree, and a threshold for fragmentation to the CreateCompleteSubtreeFragments procedure and receives CompleteSubtree and SubgroupRange for each GKB fragment. If there is only one GKB fragment created, SubgroupRange is removed.

3) If ComplementSubtreeFlag = 1,

i) If CreateCompleteSubtree procedure is used, the MIS user sends leaf numbers that do not correspond with MISF IDs of the group members and all Node Indices which represent the group management tree to the CreateCompleteSubtree procedure and receives CompleteSubtree for the GKB.

ii) If CreateCompleteSubtreeFragments procedure is used, the MIS user sends leaf numbers that do not correspond with the MISF IDs of the group members and all Node Indices which represent the group management tree, and a threshold for fragmentation to CreateCompleteSubtreeFragments procedure, and receives CompleteSubtree and SubgroupRange for each GKB fragment. If there is only one GKB fragment created, SubgroupRange is removed.

e) (Optional) Generate GroupKeyData, MasterGroupKey, VerifyGroupCode and set CompleteSubtree:

1) When MGK is not distributed, this process is skipped.

2) Send the MGK and the CompleteSubtree to the MasterGroupKeyWrapping procedure, and receive GroupKeyData. The procedure accesses the *Tree Information Base* to refer all the pairs of a Node Index and a corresponding node key.

3) (Optional) Set MGK to MasterGroupKey.

4) (Optional) Send the MGK to the verify group code generating procedure, and receive VerifyGroupCode.

5) (Optional) Set CompleteSubree to Bloom Filter of a list of Node Indices in the CompleteSubtree.

6) (Optional) Delete CompleteSubtree.

f) (Optional) Construct the UserSpecificData field.

g) Choose a DestinationIdentifier. A DestinationIdentifier is an MISF Group ID, which represents an existing group. The group indicated by the DestinationIdentifier shall include all recipients who are manipulated by this command. (The group indicated by MIS Broadcast ID always includes all recipients.)

h) If a response message is not required, ResponseFlag is set to ‘0’. If a response message is required, ResponseFlag is set to ‘1’.

i) Generate an MIS_Push_Group_Manipulate.request from the DestinationIdentifier, the TargetIdentifier, the SubgroupRange (optional), the UserSpecificData (optional), the CompleteSubtree, ComplementSubtreeFlag (optional), ResponseFlag (optional), GroupKeyUpdateFlag, MasterGroupKey (optional) and the GroupKeyData (optional). Set the GroupKeyUpdateFlag if the MGK of the group designated by the TargetIdentifier should be updated. Send it to the local MISF.

j) Optionally, in case the MIS user of PoS with group manager obtains a Transport Address to be used by the group (through any means outside of this specification), it can choose to ask the MISF to use it by including it in the MIS_Push_Group_Manipulate.request.

NOTE—Steps e) through j) are performed for each GKB fragment.

Figure 68 and Figure 69 show a flow diagram summarizing the steps performed by the MIS user on a PoS, described in this Clause. Figure 70 shows a flow diagram summarizing the steps to define CompleteSubtree and SubgroupRange which are corresponding with CreateCompleteSubtreeFragments procedure in Figure 68 and Figure 69.

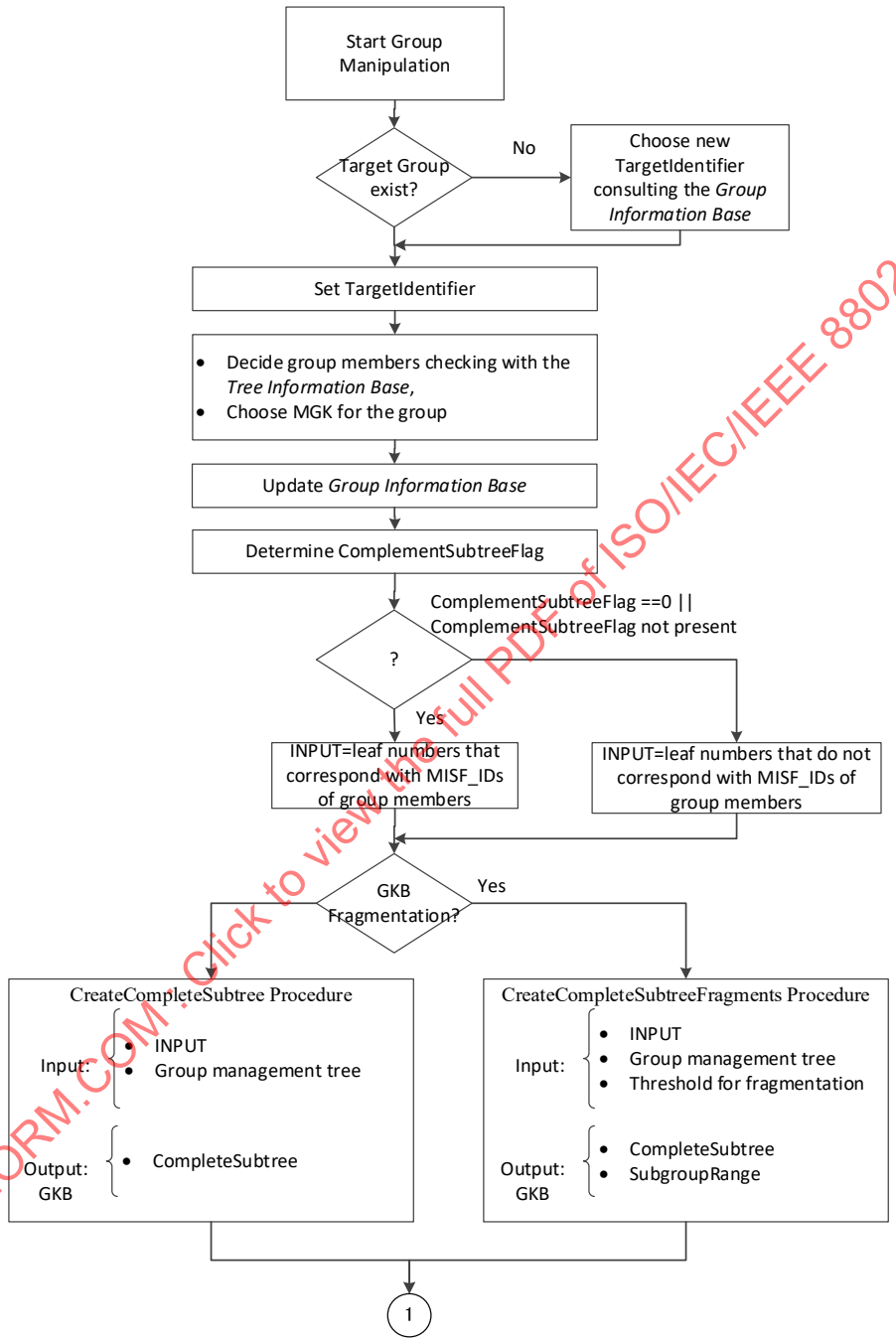


Figure 68—Summary of steps performed by MIS user of PoS with group manager

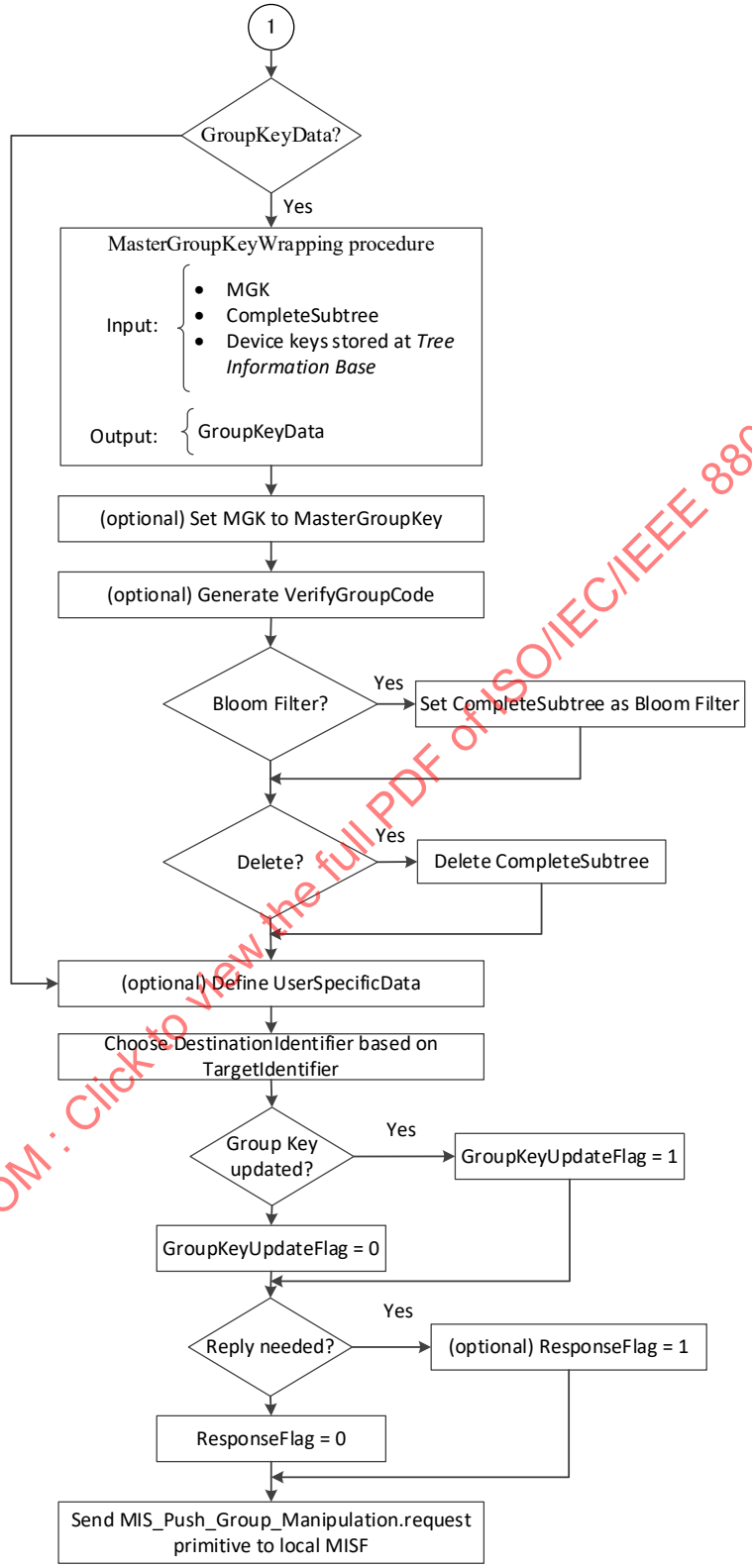


Figure 69—Summary of steps performed by MIS user of PoS with group manager
(continued from Figure 68)

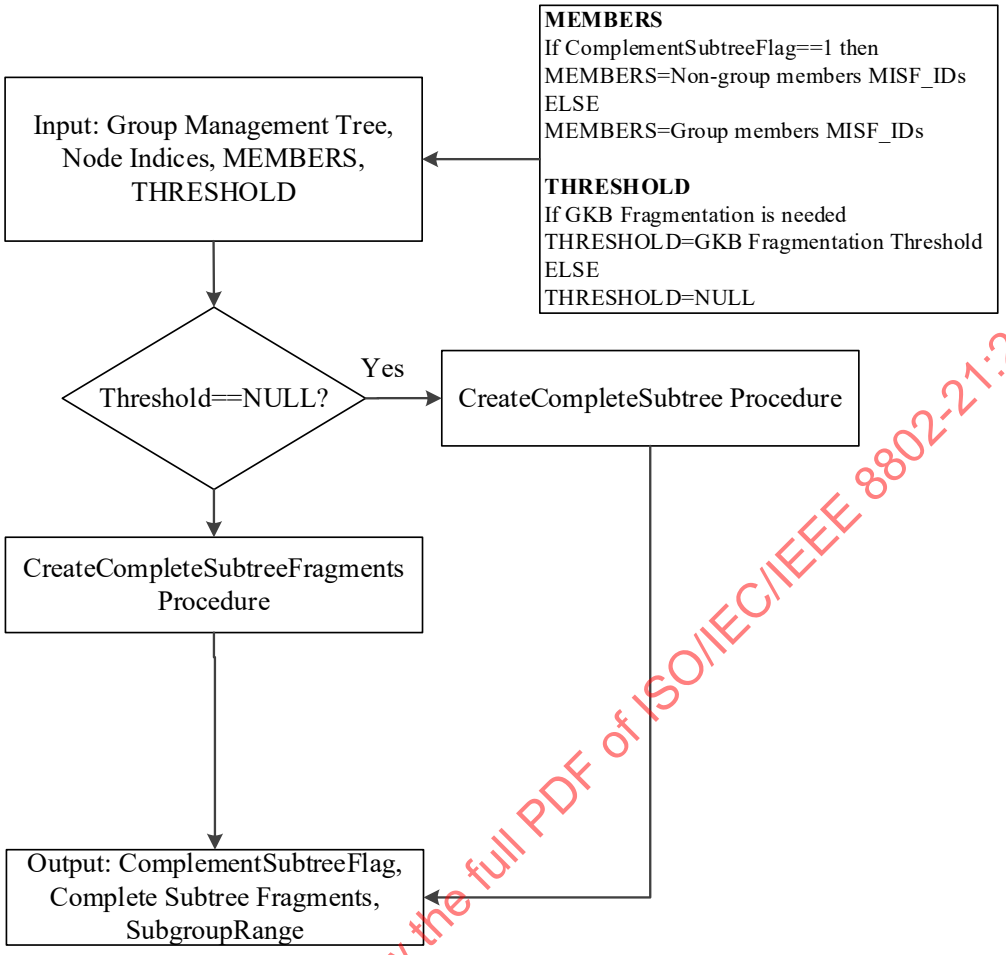


Figure 70—Flow diagram of CreateCompleteSubtree and CreateCompleteSubtreeFragments procedure

9.5.3.1.2 MISF of a PoS with group manager

Required components relevant to group manipulation and group commands are listed as follows:

— A signing key (of type SIGNING_KEY as defined in Table E.25). The key is for creation of a signature at the PoS with group manager.

— A *Group Membership Information Base* (of type GROUP_MEMBERSHIP_BASE as defined in Table E.25) stores the information required to send and receive commands to/from the group, i.e., the MISF Group ID and the transport address used. It also stores a sequence number for outgoing messages, a list of pairs of MISF ID and sequence number for incoming messages. If the service specific TLVs, which are carried in group addressed commands and group manipulation commands addressed to the group are encrypted, the *Group Membership Information Base* also stores the MGK and the SAID associated with the group.

Allocation of a transport address to an MISF Group ID is implementation specific and outside the scope of this standard. The transport address may be contained in the MIS_Push_Group_Manipulate.request received from the MIS user. When the MISF receives an MIS_Push_Group_Manipulate.request, generated by the MIS user, the MISF generates and sends an MIS_Push_Group_Manipulate indication/request message to an MISF Group or an MISF. Note that this behavior depends on the ResponseFlag parameter.

When “ResponseFlag = 1,” the MISF generates MIS_Push_Group_Manipulate request message. When “ResponseFlag = 0,” the MISF generates MIS_Push_Group_Manipulate indication message.

The following details the steps performed to generate the MIS service specific TLVs of the message:

a) Generate a group identifier TLV from the TargetIdentifier in the received MIS_Push_Group_Manipulate.request.

b) If the MIS_Push_Group_Manipulate.request contains a SubgroupRange, it generates a SubgroupRange TLV from the SubgroupRange.

c) If the MIS_Push_Group_Manipulate.request contains a UserSpecificData, it generates an aux data TLV from the UserSpecificData.

d) Generate a complete subtree TLV from the CompleteSubtree in the received MIS_Push_Group_Manipulate.request.

e) If the MIS_Push_Group_Manipulate.request contains a GroupKeyData, it generates a group key data TLV from the GroupKeyData.

f) If MIS_Push_Group_Manipulate.request contains a ComplementSubtreeFlag, it generates complement subtree flag TLV from the ComplementSubtreeFlag.

g) If the MIS_Push_Group_Manipulate.request contains a VerifyGroupCode, it generates a verify group code TLV from the VerifyGroupCode.

h) The MISF generates a sequence number TLV.

i) The MISF generates optionally a transport address TLV. If the MIS_Push_Group_Manipulate.request contains a TransportAddress parameter, the parameter is contained in the transport address TLV. Else if the MIS_Push_Group_Manipulate.request does not contain a TransportAddress parameter, the MISF decides a transport address parameter (which is implementation specific and outside of the scope of this specification).

j) If GroupKeyData is accompanied, generate an SAID notification TLV. If GroupKeyUpdateFlag = 0, the TLV contains the security association identifier associated with the GroupKeyData. Otherwise, the TLV contains a newly allocated security association ID for the GroupKeyData. The security association identifier obtained through the SAID notification TLV is stored. This identifier is used in SAID TLVs, which are carried in subsequent MIS messages encrypted by the group key corresponding to the GroupKeyData.

k) Update the *Group Membership Information Base* with TargetIdentifier, the transport address parameter, the sequence number and the SAID. If a MasterGroupKey is contained in the MIS_Push_Group_Manipulate.request, also update the *Group Membership Information Base* with the MasterGroupKey as the MGK.

Figure 71 shows a flow diagram summarizing the steps performed by the MISF at a PoS, described in this Clause.

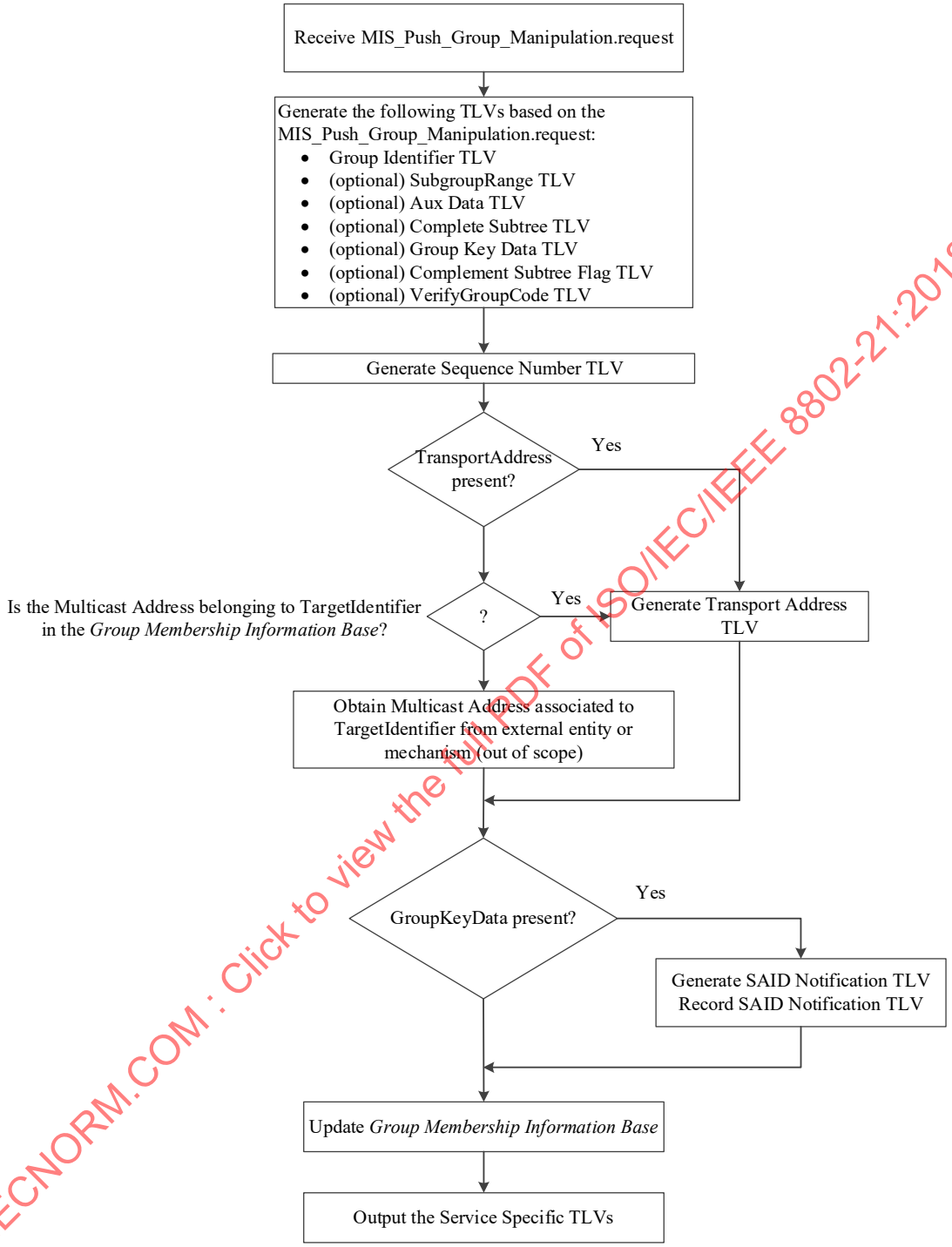


Figure 71—Summary of steps performed by MISF of PoS with group manager

9.5.3.2 Receiving procedures for group manipulation commands

The recipient of a group manipulation message is either an MN or a PoS that receives and understands group manipulation commands.

Required components relevant to group manipulation and group commands are listed as follows:

— A *Group Recipient Information Base* (of type GROUP_RECIPIENT_INFO_BASE as defined in Table E.25) containing the pairs of a Node Index and a corresponding node key (i.e., device keys) to retrieve an MGK from a GKB and the certificate used to verify digital signatures.

— A *Group Membership Information Base* (of type GROUP_MEMBERSHIP_BASE as defined in Table E.25) stores the information required to receive and response commands to the group, i.e., the MISF Group ID and the transport address used. It also stores a sequence number for outgoing messages, a list of pairs of MISF ID and sequence number for incoming messages. If the service specific TLVs carried in group addressed commands and group manipulation commands addressed to the group are encrypted, the *Group Membership Information Base* also stores the MGK and the SAID associated with the group.

When an MISF receives an MIS_Push_Group_Manipulate indication/request or MIS_Pull_Group_Manipulate response message, it processes the message, after reassembling fragments if any, as follows:

a) If a SubgroupRange TLV does not exist in the message, go to Step b). Otherwise, the MISF obtains a SubgroupRange and checks whether its own Leaf Number is contained in the SubgroupRange or not. If it is not, the MISF shall cancel the following steps and stop processing.

b) The MISF processes the Complete Subtree in the complete subtree TLV, a GroupKeyData in the group key data TLV, and a VerifyGroupCode in the verify group code TLV as described in 9.5.2.2. If the MISF succeeds to find a matching pair of Node Indices, go to Step c). Otherwise, go to Step d).

c) If ComplementSubtreeFlag in the complement subtree flag TLV is ‘0’ or the complement subtree flag TLV is not present, go to Step e), else go to Step m).

d) If ComplementSubtreeFlag in the complement subtree flag TLV is ‘0’ or the complement subtree flag TLV is not present, go to Step m), else go to Step f).

e) If a group key MGK is derived in Step b), the MISF obtains a SAID in the SAID notification TLV.

f) If a transport address TLV exists in the message, the MISF obtains a TransportAddress. Otherwise, the MISF obtains a transport address with respect to the TargetIdentifier from a server (note that this operation is out of the scope of this specification).

g) If a sequence number TLV exists in the message, the MISF obtains a SequenceNumber.

h) The MISF checks whether a TargetIdentifier in the group identifier TLV has already been registered or not in the *Group Recipient Information Base*. If it has been, go to Step i) [Stay]. Otherwise, go to Step k) [Join].

i) [Stay] The MISF updates the transport address, the group key and the SAID, and the SequenceNumber, with respect to the TargetIdentifier, in the *Group Membership Information Base*.

j) The MISF throws an MIS_Push_Group_Manipulate.indication described in 7.4.21.2 to the MIS user. The GroupStatus field of the indication shall be “Unchanged successful” (5). The procedure of command processing terminates.

k) [Join] The MISF starts listening to the transport address associated with the TargetIdentifier. The MISF saves in the *Group Membership Information Base* the TargetIdentifier, the associated transport address, the group key (Option), the SequenceNumber (Option), and the SAID (Option).

l) The MISF issues an MIS_Push_Group_Manipulate.indication described in 7.4.21.2 to the MIS user. The GroupStatus field shall be “Join operation successful” (0). The procedure of command processing terminates.

m) The MISF checks whether a TargetIdentifier in the Group Identifier TLV has already been registered or not in the *Group Membership Information Base*. If it has been, go to Step n) [Leave]. Otherwise, the MISF terminates the procedure of command processing.

n) [Leave] The MISF finds the transport address recorded on the same row as the TargetIdentifier and the MISF stops listening to it. The MISF removes the row that has the TargetIdentifier.

o) The MISF throws an MIS_Push_Group_Manipulate.indication described in 7.4.21.2 to the MIS user. The GroupStatus field shall be “Leave operation successful” (3). The procedure of command processing terminates.

Figure 72 summarizes the steps followed by the MISF on the MN/PoS upon reception of an MIS_Push_Group_Manipulate indication or request message, after reassembling fragments if any.

Subclause 7.4.20 introduces a mechanism enabling the recipient to trigger the Join or Leave operations controlled by the PoS with group manager. In order to do so, the MIS user located at the recipient notifies the PoS with group manager of its desire to Join or Leave a group through the use of the MIS_Pull_Group_Manipulate primitive. The MISF of the PoS with group manager, upon receiving the associated request message, performs the same process as defined in this Clause, for the use of the MIS_Push_Group_Manipulate, although in this case, the group to be manipulated is provided by the recipient. The resulting GKB parameters are returned to the recipient in the MIS_Pull_Group_Manipulate response message.

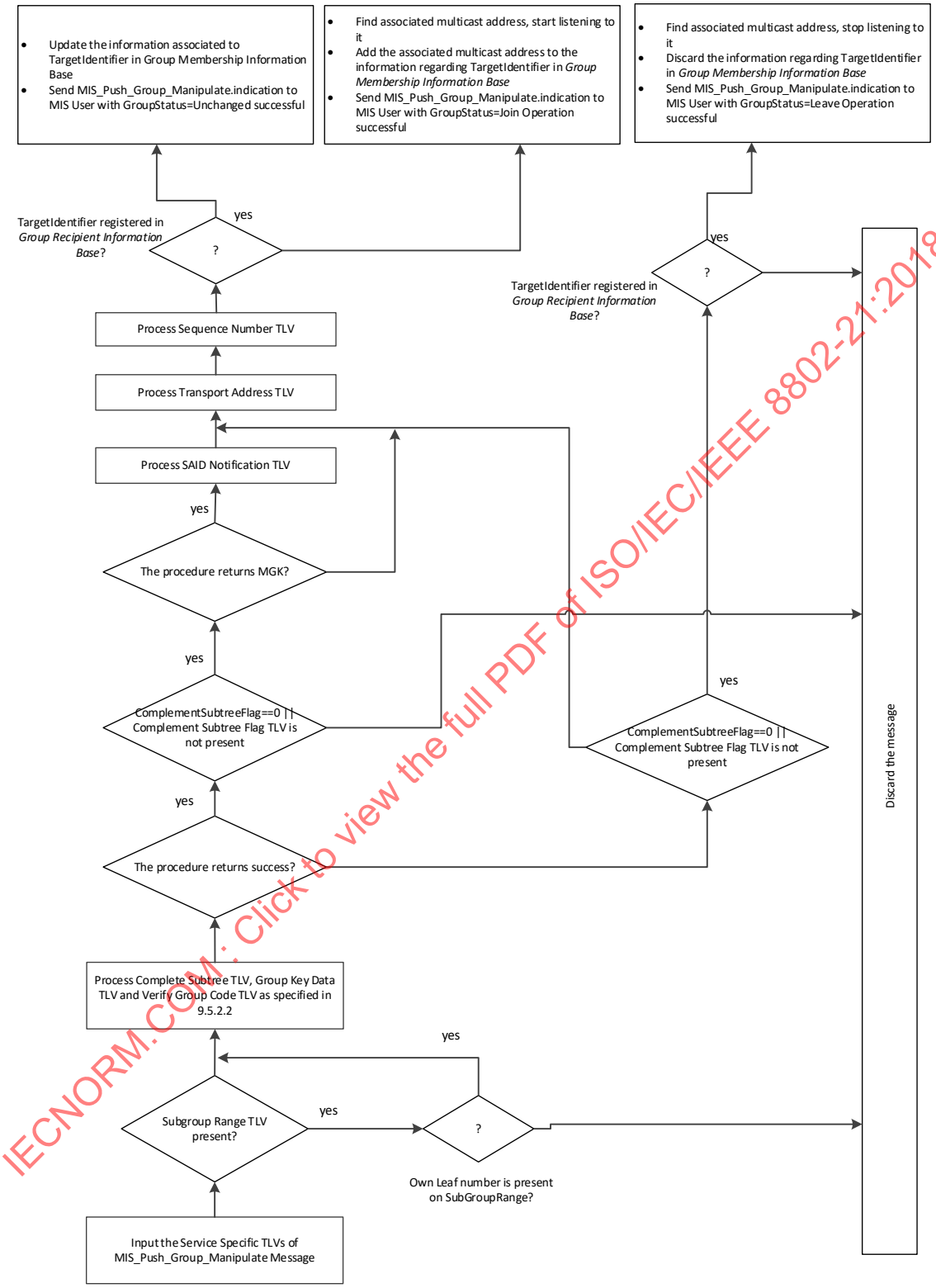


Figure 72—Summary of steps performed by the recipient MISF

9.6 Group addressed message protection

9.6.1 Group session key derivation

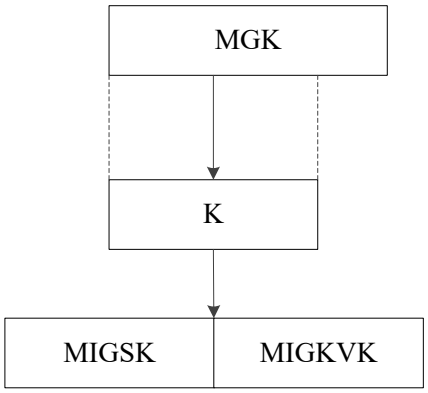


Figure 73—Key derivation example

When a recipient of a GKB successfully decrypts an MGK from the GKB, a media independent group session key (MIGSK) is derived from the MGK to protect group manipulation commands and group addressed commands:

For the key derivation, the following notations and parameters are used:

- *K*: key derivation key. It is truncated from a master group key (MGK). The length of *K* is determined by the pseudorandom function (PRF) used for key derivation. If HMAC-SHA-1 or HMAC-SHA-256 is used as a PRF, then the full MGK is used as key derivation key, *K*. If CMAC-AES is used as a PRF, then the first 128 bits of MGK are used as derivation key, *K*.
- *L*: The binary length of derived keying material MIGSK and MIGKVK. *L* is determined by selected group ciphersuite described in 9.6.5.
- *h*: The output binary length of PRF used in the key derivation. That is, *h* is the length of the block of the keying material derived by one PRF execution. Specifically, for HMAC-SHA-1, $h = 160$ bits; for HMAC-SHA-256, $h = 256$ bits; for CMAC-AES, $h = 128$ bits.
- *n*: The number of iterations of PRF in order to generate *L*-bits keying material.
- *c*: The group ciphersuite code is a one octet string specified for each ciphersuite. The code is defined in 9.6.5.
- *v*: The length of the binary representation of the counter and the length of keying material *L*. The default value for *v* is 32.
- "MIGSK": 0x4D4947534B, ASCII code in hex for string "MIGSK."
- $[a]_2$: Binary representation of integer *a* with a given length.

For given PRF, the key derivation for MIGSK and MIGKVK can be described in the following procedures:

Fixed input values: h and v.

Input: *K*, *L*, and group ciphersuite code.

Process:

a) $n := \lceil L/h \rceil$
b) If $n > 2^v - 1$, then indicate an error and stop.

c) Result(0) := empty string.
d) For i = 1 to *n*, do
   1) $K(i) := PRF(K, \text{“MIGSK”} \| [i]_2 \| c \| [L]_2)$.
   2) Result(i) = Result(i − 1) || *K*(i).
e) Return Result(n) and MIGSK is the leftmost *L* bits of Result(n).

Output: MIGSK || MIGKVK.

With the above procedure, a key hierarchy is derived as shown in Figure 73.

This mechanism conforms with NIST SP800-108 (KDF in Counter Mode).

9.6.2 Multicast message protection

Depending on the selected group ciphersuite, an MIS PDU should be encrypted, integrity protected, or protected in both aspects. An example procedure is illustrated in Figure 74.

In order to send a group addressed message the MIS user of the PoS with group manager generates a request primitive and delivers it to the local MISF. Upon receiving the request, the MISF behaves as follows:

a) The MISF generates an MIS request or indication message contained an MIS header, a source MISF identifier TLV, a destination MISF identifier TLV, and service specific TLVs or a fragment.
b) Consulting with the *Group Membership Information Base*, the MISF finds the transport address for the group associated with the DestinationIdentifier in the received request.
c) The MISF runs PDU protection procedure in Figure 74.
   1) The MIS service specific TLVs or a fragment should be encrypted with an MIGSK associated to the DestinationIdentifier to make a security TLV if necessary in the scheme described in 9.6.3.
   2) A signature TLV should be generated as shown in 9.6.4.1 using the signing key of the MISF.
d) The MISF sends the message to the transport address found in Step c)

The MIS service specific TLVs or fragment should be encrypted to make a security TLV if necessary in the scheme described in 9.6.3.

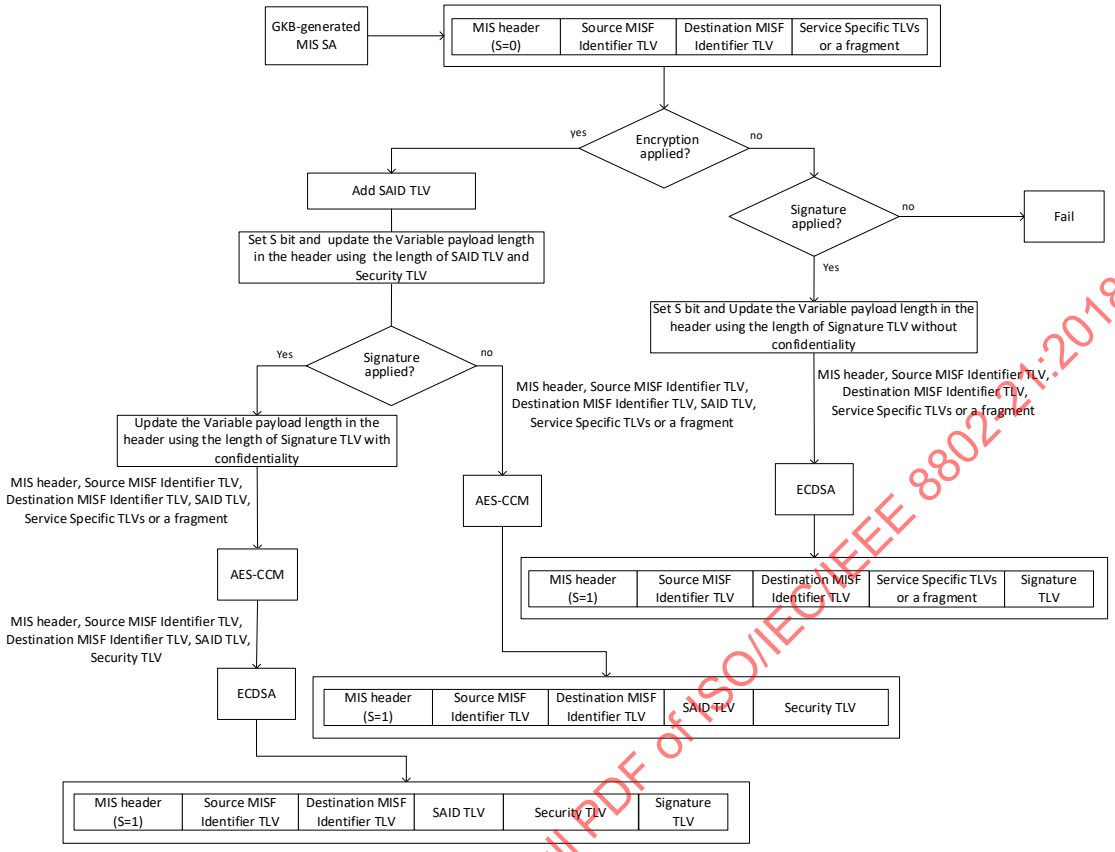


Figure 74—MIS PDU protection procedure with the GKB-generated MIS SA

When an MISF of a recipient receives the message, the following steps are taken:

a) The Destination Identifier is retrieved from the destination MISF ID TLV. The MISF checks if the Destination Identifier is registered in the Group Information Base or not. If it is not, the message is not for the recipient. Thus, it cancels the following steps and stops processing.

b) The Source Identifier is retrieved from the source MISF ID TLV.

c) If the signature TLV is attached, the MISF verifies SIGNATURE_DATA in the signature TLV using the verification key corresponding with the CERT_SERIAL_NUMBER in the signature TLV as shown in 9.6.4.2.

d) If a security TLV is contained, the MISF decrypts the security TLV in the scheme described in 9.6.3 with the MIGSK associated with the SAID in the security TLV that is available in the Group Information Base. If the decryption fails, it cancels the following steps and aborts. If the decryption succeeds, MIS service specific TLVs or a fragment is obtained.

e) If MIS service specific TLVs is obtained or a fragment is obtained and reassembling fragments succeeds, the MISF issues an indication primitive to its local MIS user.

9.6.3 MIS PDU protection for group addressed message by AES-CCM

The parameters used in AES-CCM, the nonce construction, the operational procedures, and the security TLV under AES-CCM protection shall be set according to the rules given in 9.6.3.1 through 9.6.3.3.

9.6.3.1 AES-CCM parameters

For AES-CCM the following parameter values shall be set:

a) *t*: The length of MIC is 12 octets (96 bits).
b) *n*: The length of the nonce *N* is 13 octets (104 bits).
c) *q*: The length of the binary representation of the octet length of the data to be encrypted is 2 octets (16 bits).
d) *a*: The length of the binary representation of the octet length of the associated data is 2 octets (16 bits).

9.6.3.2 Construct AES-CCM nonce

AES-CCM uses a nonce to construct an initialization vector and the counter. CCM requires a unique nonce value for each MIS message protected by a given MIGSK as described in 9.3.3.2.

The SN shall never be repeated for a series of encrypted MIS PDUs using the same MIGSK.

9.6.3.3 Operational procedures in AES-CCM

9.6.3.3.1 Encapsulation

For a given GKB-generated MIS SA, the prerequisites for AES-CCM encapsulation includes an encryption key MIGSK, an AES block cipher encryption block, and the values of parameters *t*, *n*, *q*, and *a*. The plaintext, *P,* to be encrypted and authenticated is formed by concatenating all the service specific TLVs as presented in MIS PDU with the padding. The associated data, *A*, is not encrypted but authenticated. *A* is formed by concatenating the MIS Header, the source MISF sdentifier TLV, the destination MISF identifier TLV as presented in MIS PDU, and the SAID TLV with padding. The data, *P* and *A,* is partitioned with necessary padding to 16-octet blocks *B1*, *B2*, ..., *Br* as specified in SP 800-38C. The octet block, *B0,* is an initialization vector and formed with 1-octet flags with associated data, 13-octet nonce *N*, and 2-octet integer *Q*, where *Q* is the octet length of *P*. The format of *B0* is illustrated in Figure 75.

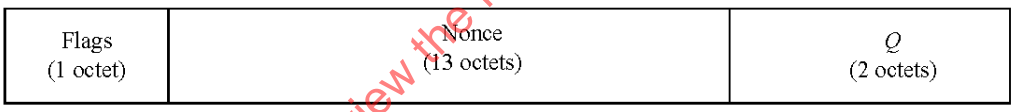


Figure 75—Format of *B0* with associated data

The flags are formed by the following data:

— 1 reserved bit, which is set to '0';
— 1 bit flag for the associated data, which is one;
— 3 bits to represent $(t - 2)/2$, which is 101 ($t = 12$);
— 3 bits to represent $q - 1$, which is 001 ($q = 2$).

The counter *Ctr(i)*, $i = 0, 1, ..., r$, is formed as described in 9.3.3.3.1.

The encapsulation of an MIS PDU consists of the following steps:

a) Fetch Transaction ID and FN from the MIS header.
b) Increment a positive number of *SN* to update the SN.
c) Construct the nonce, *N*, as described in 9.3.3.2.
d) Input *N, A,* and *P* to AES-CCM generation-encryption process as specified in SP 800-38C. The *B0* and all the counter numbers are formed as described in Figure 75 and Figure 50, respectively.
e) Obtain the output, C, of AES-CCM.

9.6.3.3.2 Decapsulation

For a given GKB-generated MIS SA, the prerequisites for AES-CCM decapsulation includes an encryption key MIGSK, AES block cipher encryption block, and the parameters *t*, *n*, *q* and *a*.

The decapsulation of a protected MIS PDU consists of the following steps:

a) Fetch Transaction ID and FN from the MIS header.
b) Fetch *SN* from the security TLV.
c) Construct the nonce, *N,* as described in 9.3.3.2.
d) Input *N, A,* and *C* to AES-CCM decryption-verification process as specified in SP 800-38C. The *B0* and all the counter numbers are formed as described in 9.6.3.3.1.
e) Obtain the output, *P,* or "INVALID."

9.6.3.4 Format of security TLV

The ENCR_BLOCK data of the security TLV in a protected MIS message with AES-CCM is described in 9.3.3.4.

9.6.4 Signature and credential management

In order to enable signing functionality, the message source requests credentials for public key using an out-of-band mechanism that is not specified in this document. The message source provides the credentials to destination devices. Message signing procedure, signature verification procedure, and certificate management procedure are described in 9.6.4.1 and 9.6.4.2, respectively.

In this specification, Elliptic Curve Digital Signature Algorithm (ECDSA) specified in IEEE Std 802.1AR-2009, Secure Device Identity, by reference to NIST FIPS 186-4 and ANSI X9.62-2005 is used as the multicast signature scheme. In particular, NIST recommended elliptic curve P-256 and hash function SHA-256, specified in FIPS 180-4, are used to generate signatures. These algorithm identifiers are defined in 9.6.4.4.

9.6.4.1 Multicast message signatures

Multicast Messages are signed with the message source using a private key of the message source. Integrity and proof of origin of a multicast message is verified by verifying the message signature with the public key of a message source. The message content is signed using elliptical curve cryptography (the signature algorithm is defined in 9.6.4).

In case the MIS PDU is protected through GKB-generated MIS SA with a signature as specified in 8.4.2.4, the MISF of PoS generates a signature TLV consisting of a CERT_SERIAL_NUMBER and a SIGNATURE_DATA. The SIGNATURE_DATA is created by signing an MIS_Group_Manipulate command or a group addressed command using a signing key corresponding with a verification key specified by CERT_SERIAL_NUMBER. Figure 76 illustrates the data protection procedure with confidentiality.

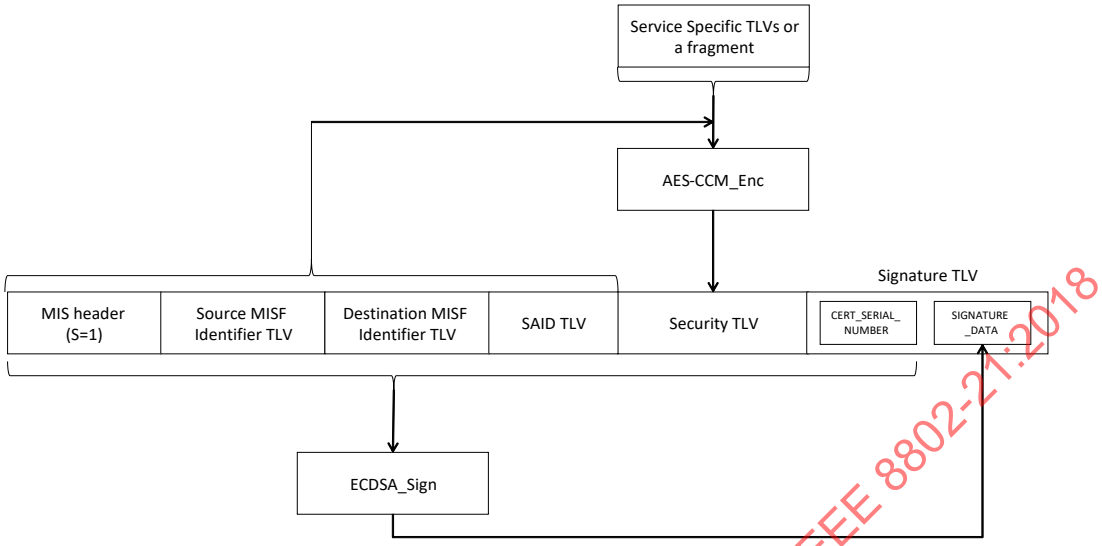


Figure 76—Signing with confidentiality

The MISF encrypts the MIS service specific TLVs or fragment with MIGSK, and generates the security TLV. The MISF selects a certification serial number and generates CERT_SERIAL_NUMBER from the certification serial number. Then, the MISF computes the SIGNATURE_DATA of the signature TLV from the MIS Header, the source MISF identifier TLV, the destination MISF identifier TLV, the SAID TLV, the security TLV, and the Type, Length, and Value fields of the signature TLV excluding the SIGNATURE_DATA.

If the MIS PDU is protected through a GKB-generated MIS SA, the signature TLV shall not include the SEQUENCE_NUMBER.

In case the MIS PDU is not protected through a GKB-generated MIS SA and protected with a signature only as specified in 8.4.2.5, the MISF of PoS generates a signature TLV consisting of a SIGNATURE_DATA, a CERT_SERIAL_NUMBER and a SEQUENCE_NUMBER. Figure 77 illustrates the data protection procedure without confidentiality.

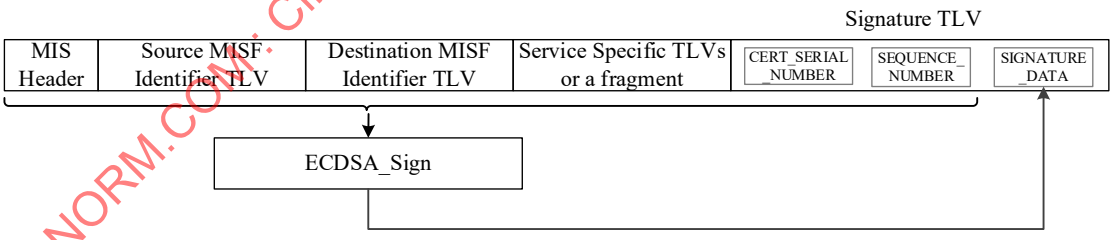


Figure 77—Signing without confidentiality

The MISF computes the SIGNATURE_DATA of the signature TLV from the MIS Header, the source MISF identifier TLV, the destination MISF identifier TLV, the service specific TLVs, and the type, length, and value fields of the signature TLV excluding the SIGNATURE_DATA.

If the MIS PDU is not protected through a GKB-generated MIS SA, the signature TLV shall include the SEQUENCE_NUMBER.

On receipt of signed multicast message there is an optional response indicating the validity of signature.

Message source requests credentials for key updates. Message source provides updates of credentials to destination devices (with overlap period).

9.6.4.2 Signature verification

The signature is verified using the message source signature verification key. The message source identifies which key is to be used for the multicast message so that verification utilizes the correct key for signature verification.

In case the MIS PDU received contains a signature TLV and is protected through a GKB-generated MIS SA, then the MISF of recipient retrieves a CERT_SERIAL_NUMBER and a SIGNATURE_DATA from the signature TLV. Then, the MISF verifies the SIGNATURE_DATA using a verification key specified by the CERT_SERIAL_NUMBER. If the signature TLV includes a SEQUENCE_NUMBER, the MIS PDU shall be dropped since it is a wrong form. Figure 78 illustrates the data protection procedure with confidentiality.

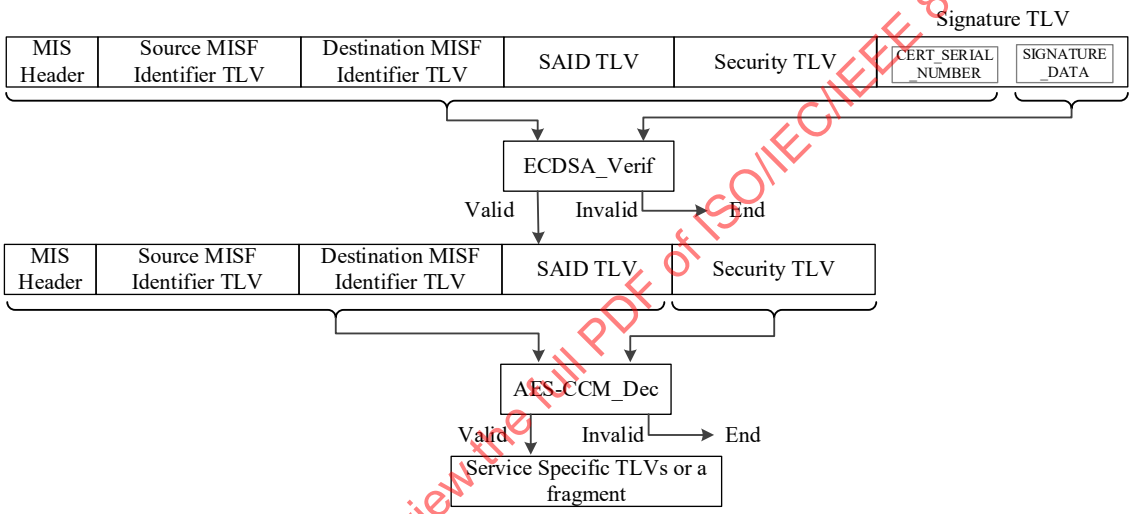


Figure 78—Signature verification with confidentiality

In the case where the MIS PDU contains a signature TLV and is not protected through a GKB-generated MIS SA, then the MISF of recipient retrieves CERT_SERIAL_NUMBER, SIGNATURE_DATA and SEQUENCE_NUMBER from the signature TLV. Then, the MISF verifies the SIGNATURE_DATA using a verification key specified by the CERT_SERIAL_NUMBER, and the SEQUENCE_NUMBER. The received SEQUENCE_NUMBER is considered valid if and only if the SEQUENCE_NUMBER is greater than the last valid incoming sequence number maintained for the sender. Figure 79 illustrates the data protection procedure without confidentiality.

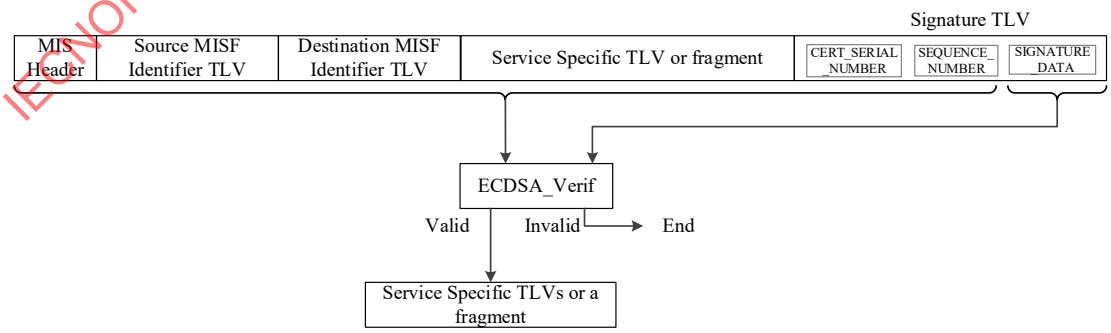


Figure 79—Signature verification without confidentiality

9.6.4.3 Certificate management

A root of trust shall exist for the multicast nodes. The root of trust is envisioned to be a certificate authority. X.509 format certificates shall be utilized. The root of trust establishes the binding between the identity of the message source and the public/private key pair used for signature generation and verification.

The certificate shall include the identity of the certificate authority, the identity of the message source, the public key in use, the expiration date of the certificate, and the certificate authority's signature. For an endpoint (an MN or PoS) to trust the certificate it shall have the certificate authority public key.

The initial certificates for multicast signature verification are distributed to multicast destinations as part of the provisioning process to the multi-node network. The certificates should include the certificate authority certificate used to verify the initial and updated certificates.

There should also be one or more certificates that are bound to the identity of the multicast source.

As part of the key update or revocation process, a new certificate should be provided to multicast destinations using the multicast mechanism. There needs to be a mechanism for multicast destinations to acknowledge the receipt of the multicast message.

When there is a suspicion that a certificate is compromised, a mechanism should be provided to revoke the certificate from service. This mechanism allows for using the multicast messaging mechanism. Multicast destinations shall provide a reply that indicates they have successfully revoked the certificate, if the message source requests the reply to be sent.

9.6.4.4 Algorithm identifiers

The ECDSA signature method is defined in IEEE Std 802.1AR-2009, Secure Device Identity, by reference to NIST FIPS 186-4 and ANSI X9.62-2005.

If implementing ECDSA P-256, the SHA-256 message digest algorithm and the P-256 elliptic curve as defined in FIPS 186-4 Annex D, D.1.2.3, shall be used.

The signature algorithm shall be ecdsa-with-SHA256 as specified in IEEE Std 802.1AR-2009 by reference to IETF RFC 5008. The object identifier is:

```
ecdsa-with-SHA256 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
ansi-X9-62(10045) signatures(4) ecdsa-with-sha2(3) 2 }
```

When the ecdsa-with-SHA256 algorithm identifier appears in the algorithm field as an AlgorithmIdentifier, the encoding shall omit the parameters field. That is, the AlgorithmIdentifier shall be a SEQUENCE of one component, the object identifier ecdsa-with-SHA256.

9.6.5 Group ciphersuites

The ciphersuites used for securing group addressed messages are defined in Table 27.

Table 27—Group ciphersuites

Code	Encryption algorithm	Digital signature algorithm
10000100	NULL	ECDSA-256
10010001	AES_CCM-128	NULL
10010101	AES_CCM-128	ECDSA-256

In Table 27, AES-CCM is an AES mode of operations specified in NIST SP 800-38C. AES-CCM provides confidentiality and data integrity.

In Table 27, ECDSA-256 uses curve P-256 and hash function SHA-256.

Notice that AES-CCM uses the group key MIGSK. It can provide data integrity but not unique data origin authentication because the symmetric key is shared among a group of recipients. The data origin authentication is provided through ECDSA.

The support of code '10000100' is mandatory and all entities supporting this specification shall implement it.

9.6.6 Group key distribution ciphersuites

The ciphersuites used for distributing a master group key are defined in Table 28.

Table 28—Group key distribution ciphersuites

Code	Wrapping algorithm	MAC algorithm for VerifyGroupCode
11010100	AES_Key_Wrapping-128	NULL
11000100	AES_ECB-128	NULL
11000101	AES_ECB-128	AES-CMAC-128
11000000	No group key distribution	NULL

In Table 28, AES_Key_Wrapping is an AES mode of operations specified in NIST SP 800-38F.

Note that ECB mode is not recommended to protect a key, because it cannot provide proper security level for the key. In particular, the same plaintext is encrypted to the same ciphertext, since no random IV is used for each encryption. On the other hand, if transmitting IVs increases the size of GroupKeyData to an unacceptable point for the transport protocol, then ECB mode may be used, assuming that the same key is retransmitted with a very small probability and signature is applied to provide authentication and integrity.

The support of code '11010100' is mandatory and all entities supporting this specification shall implement it. The default PRF used for key derivation function specified in 9.6.1 is PRF_CMAC_AES.

Note that digital signature algorithm ECDSA-256 is used to protect group key distribution.

10. Proactive authentication

In a handover from a service PoA to a target PoA, a mobile node may need to authenticate to the target network through an authentication mechanism required by the target network. This clause specifies the mechanisms to use MIS to assist proactive authentications to reduce the latency due to media access authentication and key establishment.

This standard introduces two options to conduct the proactive authentication with a targeted network. The first option is called unbundled media access proactive authentication. In such a proactive authentication, an MN conducts an authentication with the targeted network as it is required for accessing that network through a specific media. In this case, the authenticator is a media-specific authenticator (MSA). The authentication messages are passed between the MN and the MSA through an MIS PoS. The authentication messages between the MN and the PoS are carried through MIS messages. The second option is to bundle the media access proactive authentication to the MIS service access authentication. In this case, at the end of MIS service access authentication, the MN and the PoS also establish a key(s) for a target PoA(s). The key(s) are distributed to the PoA(s) so that when a handover to one of the PoAs happens, the MN can

establish a protected link with the PoA. The MIS message exchange between an MN and a PoS is common to both bundled and unbundled proactive authentication. The only difference is that the bundled proactive authentication uses a key established through MIS service access authentication. The MIS message exchange for bundled and unbundled proactive authentication is described in 10.1.

10.1 Media-specific proactive authentication

In a media access proactive authentication, a target PoS passes authentication messages between the mobile node and a media-specific authenticator (MSA). The protocol stacks in each interface are illustrated in Figure 80 and Figure 81. In scenarios where MSA/target PoA is reachable via same media as MN and PoS, EAP messages received at the PoS are directly forwarded to the target PoA. In an optimized pull key distribution, SPoS passes authentication messages between the mobile node, the target PoS, and a media-specific authenticator (MSA).

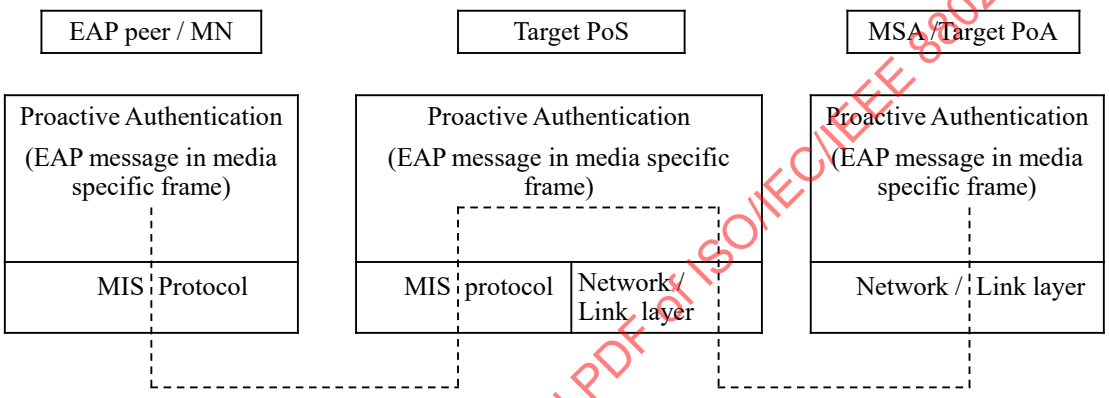


Figure 80—Protocol stack for MIS supported proactive authentication

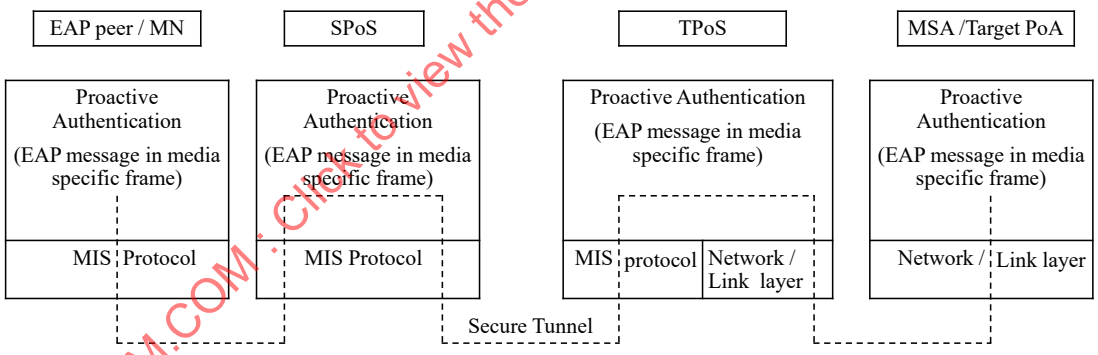


Figure 81—Protocol stack for MIS supported optimized pull key distribution with two points of service

Figure 81 illustrates the protocol stacks and message passing when the SPoS is in a different network from that of the target point of service (TPoS).

10.1.1 Procedures in a media-specific proactive authentication

An MIS assisted media-specific proactive authentication includes the following main procedures.

10.1.1.1 PoS and candidate media-specific authenticator discovery

Before an MN initiates an MIS assisted proactive authentication, the MN needs to know the PoS's address and the candidate media-specific authenticators' link layer addresses. The corresponding candidate MSAs' addresses can be discovered by using the information elements (IEs) specified in 6.5.4.

10.1.1.2 Proactive authentication through EAP or ERP

In order to execute a proactive authentication, the EAP or ERP messages are encapsulated in the extended MIS messages as L2 frames. When the PoS receives an encapsulated EAP or ERP message, it decapsulates it, then forwards it to the candidate media-specific authenticator (MSA). The EAP or ERP messages are encoded as OCTET_STRING.

10.1.1.3 Media-specific association handshake

When the MN decides to handover to a candidate network, the MN and the PoA, which is associated with the MSA, perform a media-specific association based on the keying material derived by the proactive authentication. For example, the media-specific handshake could be a 4-way handshake as in an IEEE 802.11 network. If required, a media-specific handshake further derives media-specific session keys to protect the communication between the MN and the PoA once it is attached to it.

10.1.2 Proactive authentication message format

When a proactive authentication is executed through EAP (IETF RFC 3748) or ERP (IETF RFC 6696), the EAP packets are carried by MIS messages. MIS primitives for the link-layer frames are defined in 7.4.18 for proactive authentication. The messages are defined in 8.6.1.18 and 8.6.1.19. The MIS messages for proactive authentication shall be protected by an MIS SA.

10.2 Bundling media access authentication with MIS service access authentication

When the trust relationship between media-specific network access provider and the MIS service provider allows, a proactive authentication can be optimized by bundling the media access authentication with an MIS service access authentication. In this case, at the end of a successful service access authentication, a PoS derives not only keys for MIS message protection as defined in 9.2.2, but also a key called media-specific root key (MSRK). This key is further used to derive a key or keys called media-specific pair-wise master keys (MSPMKs) to be used by a target PoA or PoAs.

10.2.1 Media-specific key derivation

10.2.1.1 Derivation of media-specific root key (MSRK)

After a successful service access authentication through EAP or ERP, a master session key (MSK) or a re-authentication MSK (rMSK) is generated in the MN and the PoS. The media-specific root key (MSRK) is derived from MSK or rMSK.

For the media-specific root key derivation, the following notations and parameters are used:

— *K*: key derivation key. It is truncated from a master session key (MSK) or re-authentication MSK (rMSK). The length of *K* is determined by the pseudorandom function (PRF) used for key derivation. If HMAC-SHA-1 or HMAC-SHA-256 is used as a PRF, then the full MSK or rMSK is used as key derivation key, *K*. If CMAC-AES is used as a PRF, then the first 128 bits of MSK or rMSK is used as key derivation key, *K*.

— *h*: The output binary length of PRF used in the key derivation. That is, *h* is the length of the block of the keying material derived by one PRF execution. Specifically, for HMAC-SHA-1, *h* = 160 bits; for HMAC-256, *h* = 256 bits; for CMAC-AES, *h* = 128 bits.
— *Nonce-T* and *Nonce-N*: The nonces exchanged during the execution of service access authentication.
— "MSRK": 0x4D53524B, ASCII code in hex for string "MSRK."

The MSRK derivation is described as follows:

Input: *K*, *Nonce-T*, *Nonce-N*.

Process:

a) MSRK:= PRF(K, "MSRK" || Nonce-T || Nonce-N).
b) Return MSRK.

Output: *MSRK*

The binary length of *MSRK* is *h*. Depending on the PRF used for the MSRK derivation, it can be 128 bits, 160 bits, or 256 bits. The MSRK is used to derive media-specific pairwise master keys (MSPMK).

10.2.1.2 Derivation of media-specific pairwise master key (MSPMKs)

Each MSPMK is derived specifically for a PoA. For the media-specific pairwise master key (MSPMK) derivation, the following notations and parameters are used:

— *K*: key derivation key. It can be a full length of *MSRK* or a portion of *MSRK*. Specifically, the length of *MSRK* is *h* which is determined by the PRF used for key derivation. If in MSRK derivation and in MSPMK derivation the same PRF is used, the MSPMK derivation is able to use the full length *MSRK*. However, in case that HMAC-SHA1 or HMAC-SHA256 is used in MSRK derivation, but CMAC-AES is used in MSPMK derivation, then only the first 128 bits of MSRK is used as a key derivation key in the MSPMK derivation.
— *MN_LINK_ID*: A link layer identity of the mobile node.
— *PoA_LINK_ID*: A link layer identity of a target point of attachment (PoA).
— "*MSPMK*": 0x4D53504D4B, ASCII code in hex for string "MSPMK."

The MSPMK derivation is described as follows:

Input: *K*, *MN_LINK_ID*, *PoA_LINK_ID*.

Process:

a) MSPMK:= PRF(K, "MS-PMK" || MN_LINK_ID|| PoA_LINK_ID).
b) Return MSPMK.

Output: *MSPMK*.

The binary length of *MSPMK* is *h*. Depending on the PRF used for the above MSPMK derivation, it can be 128 bits, 160 bits, or 256 bits.

The new key hierarchy is illustrated in Figure 82.

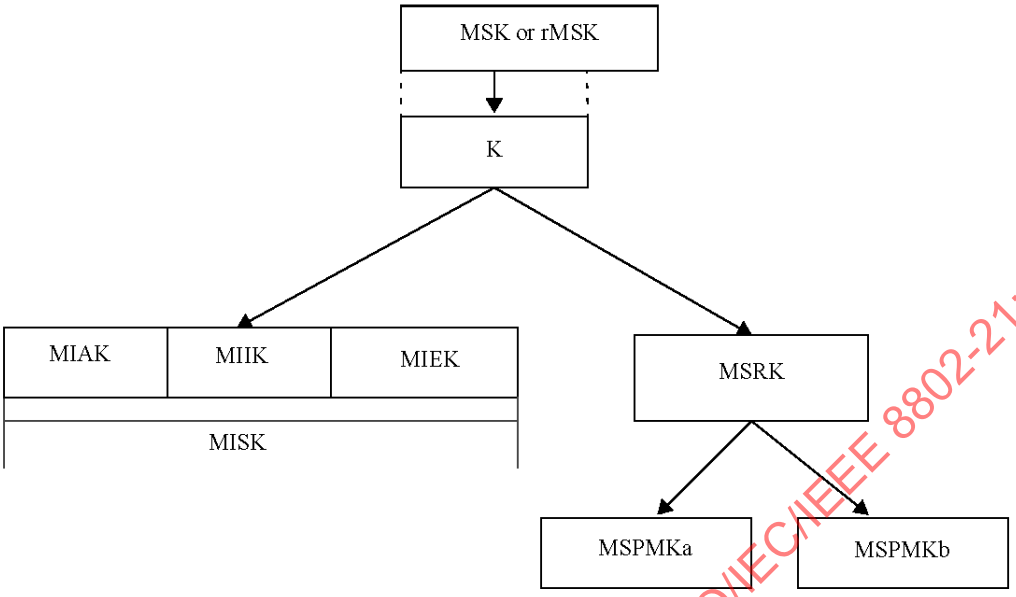


Figure 82—Key hierarchy for bundle case

10.2.2 Media-specific key distribution

Each MSPMK is distributed to a PoA. The key distribution from the PoS to a PoA can be done through push or pull key distribution. In general, key distribution from a PoS to a PoA is out of the scope of this standard. However, MIS service can be used to trigger the key distribution. The key distribution can be triggered in the following methods.

10.2.2.1 Push key distribution

The objective of push key distribution is to trigger a PoS to push a key into a target PoA. To perform the installation, the MN uses the MIS protocol, which at this point could be protected, to notify the PoS to start the key installation. In the PoS, the key is pushed from MISF to an MIS user for the further distribution to a PoA. The primitives for push key distribution are defined in 7.4.17. The messages are defined in 8.6.1.16 and 8.6.1.17.

10.2.2.2 Reactive pull key distribution

A reactive pull key distribution is performed after the MN moves to the target PoA. Since no MIS function is used, this is out of the scope of this standard.

10.2.2.3 Optimized proactive pull key distribution

This mechanism allows the MN to perform a media-specific authentication proactively with a target PoA without being directly connected to the wireless link of the target PoA by means of sending link-layer frames through the PoS to the target PoA. The key hierarchy shared between the MN and the PoS is used in order to derive a pre-shared key to conduct a proactive authentication. The PoS is acting as a local authentication server (AS). The PoA receiving the link-layer frames with the authentication information can contact with the AS (the PoS) using the identifier provided during the service access authentication. Once the proactive authentication is completed, a media-specific master session key (MSK) is distributed from the PoS (acting as an AS) to the PoA. At the end, the MN and the PoA share the same media-specific MSK. To perform this key distribution mechanism, the primitives are defined in 7.4.18 and MIS messages are defined in 8.6.1.18 and 8.6.1.19.

Annex A

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] ANSI X9.62-2005, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA).[15]

[B2] Bloom, B. H., "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM,* Vol. 13, No. 7, July 1970.

[B3] IEEE Std 100™, *The Authoritative Dictionary of IEEE Standards Terms,* Seventh Edition.[16]

[B4] IETF RFC 791 (1981-09), DARPA Internet Program Protocol.

[B5] IETF RFC 3629 (2003-11), UTF-8, A Transformation Format of ISO 10646.

[B6] IETF RFC 4291 (2006-02), IP Version 6 Addressing Architecture.

[B7] IETF RFC 5480 (2009-03), Elliptic Curve Cryptography Subject Public Key Information.

[B8] IETF RFC 5580 (2009-08), Carrying Location Objects in RADIUS and Diameter.

[B9] IETF RFC 6318 (2011-06), Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME).

[B10] ITU-T Recommendation X.210 (11/93), Information Technology-Open Systems Interconnection-Basic Reference Model: Conventions for the Definition of OSI Services [common text with ISO/IEC 10731].[17]

[B11] ITU-T Recommendation X.690, Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

[B12] ITU-T Recommendation Y.1541, Network Performance Objectives for IP-Based Services.

[B13] ITU-T Recommendation Z.120 (2004), Languages and General Software Aspects for Telecommunication Systems, Formal Description Techniques (FDT)—Message Sequence Chart (MSC).

[B14] NIST FIBS 186-4, Digital Signature Standards (DSS), July 2013.[18]

[B15] W3C Recommendation, OWL Web Ontology Language Reference.

[B16] W3C Recommendation, RDF Vocabulary Description Language 1.0: RDF Schema.

[B17] W3C Recommendation, SPARQL Query Results XML Format.

[15] ANSI publications are available from the American National Standards Institute (http://www.ansi.org/).

[16] IEEE publications are available from the Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

[17] ITU-T publications are available from the International Telecommunications Union (http://www.itu.int/).

[18] NIST publications are available from the National Institute of Standards and Technology (http://www.nist.gov/).

Annex B

(normative)

Quality of service mapping

This annex provides the mapping between quality of service (QoS) parameters with various technologies. A flow diagram is provided that shows the setting and reporting of QoS parameters using the standard IEEE 802.21 primitives. Table B.1, Table B.2, and Table B.3 show the mapping between generic QoS parameters and those used by different technologies such as IEEE 802.11, IEEE 802.16, and 3GPP. Clause B.3 describes how the generic QoS parameters can be derived from the access link specific parameters.

A transmitted packet over a communication medium can experience the following outcomes:

— Be received with no errors at its intended destination
— Be received with errors at its intended destination
— Not be received in which case it is said that the packet is lost

A communication medium represents one or multiple point-to-point network segments that are termed *links* in this standard.

The maximum attainable speed of information transfer over a given communication channel can be constant, as is usually the case with communication channels involving only wired links, or it can be time varying at different scales, as is often the case for communication channels involving wireless links. This measure is called *link throughput*, for the purposes of this standard.

The ability of the link to provide accurate information transfer can be described via a statistical model characterized by the following parameters:

— Minimum Packet Transfer Delay: is defined as the minimum delay over a population of interest.
— Average Packet Transfer Delay: is defined as the arithmetic mean of the delay over a population of interest.
— Maximum Packet Transfer Delay: is defined as the maximum delay over a population of interest.
— Jitter: is defined as the standard deviation of the delay over a population of interest.
— Packet Loss Rate: is defined as the ratio between the number of frames that are transmitted but not received and the total number of frames transmitted over a population of interest.
— Packet Error Rate: is defined as the ratio between the number of packets that have been received with errors and the total number of packets present in a population of interest. Note that if the link supports retransmission, then the Packet Error Rate includes it, otherwise it does not include it.

For a link that supports class of service (CoS) differentiation, CoS traffic accuracy parameters need to be maintained in order to provide insights on how individual traffic classes are faring.

In summary, the following set of parameters characterizes the speed and accuracy of the information transfer that a multi-CoS traffic link supports:

a) Link Throughput: the number of bits successfully received divided by the time it took to transmit them over the medium.

b) Link Packet Error Rate: represents the ratio between the number of frames received in error and the total number of frames transmitted in a link population of interest.

c) Supported Classes of Service: represents the maximum number of differentiable classes of service supported by this link.

d) Class of Service Parameters List: for each of the supported classes of service the following parameters are defined:

   1) Class Minimum Packet Transfer Delay: is defined as the minimum delay over a class population of interest.
   2) Class Average Packet Transfer Delay: is defined as the arithmetic mean of the delay over a class population of interest.
   3) Class Maximum Packet Transfer Delay: is defined as the maximum delay over a class population of interest.
   4) Class Packet Delay Jitter: is defined as the standard deviation of the delay over a class population of interest.
   5) Class Packet Loss Rate: is defined as the ratio between the number of frames that are transmitted but not received and the total number of frames transmitted over a class population of interest.

B.1 Generic IEEE 802.21 QoS flow diagram

Figure B.1 represents an example flow diagram for using the QoS framework defined by the media independent services framework (MISF).

The following terms are used in Figure B.1:

— UP Entity: An upper layer entity such as a multimedia application;
— MAC-S: The medium access control (MAC) layer of the interface that is currently serving the mobile node (MN);
— MAC-C: The MAC layer of an interface that is not currently serving the MN;
— PoA-S: The serving point of attachment (PoA);
— PoS-S: The serving PoS.

The MIS_Link_Configure_Thresholds primitive is used to set the application quality of service requirements and make it available to the MISF. These parameters are mapped into media-specific measurements at the MIS layer and then used to configure the link parameter thresholds. While this mapping is not defined by other standards, Table B.1 and Table B.2 provide such mappings. The MIS_Link_Parameters_Report primitive is used to relay link specific measurements back to the MIS user.

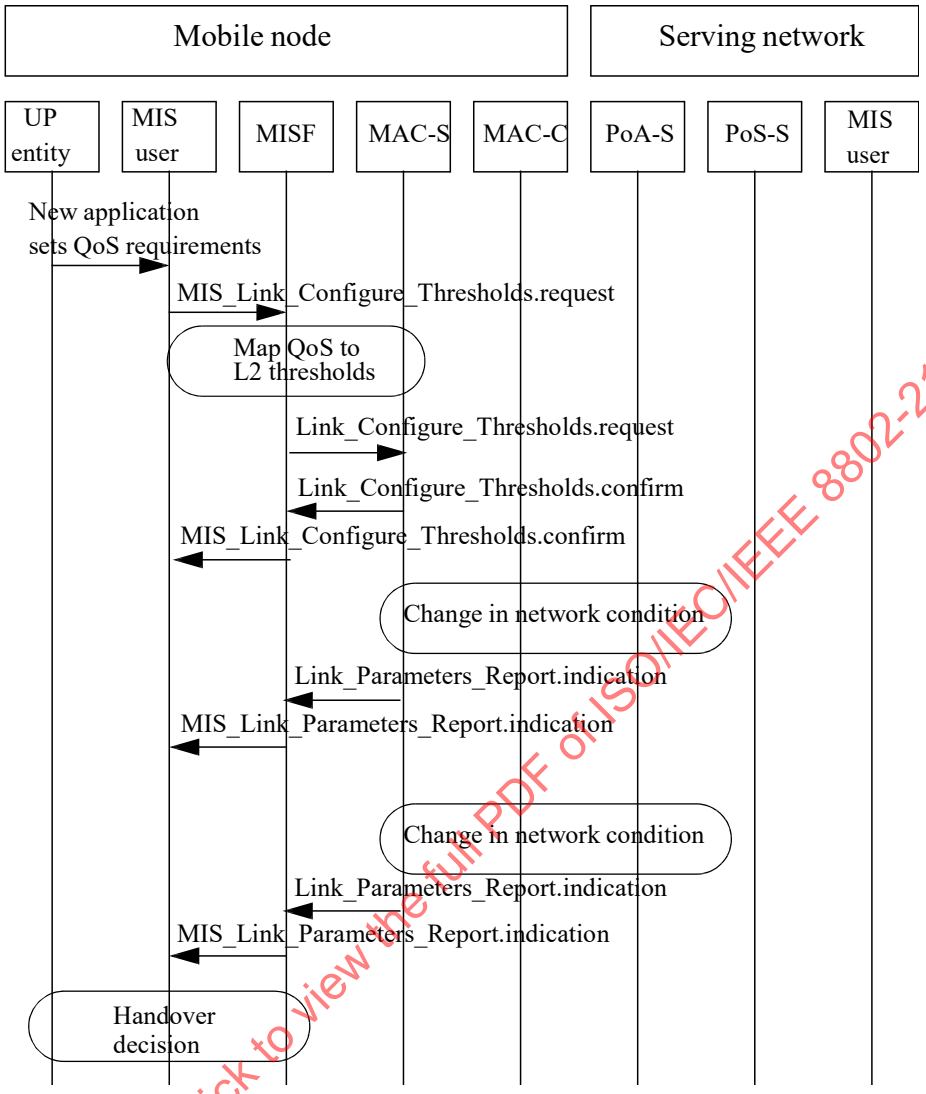


Figure B.1—An example flow for setting application QoS requirements

B.2 Generic IEEE 802.21 QoS parameter mappings

The tables provide mappings of the standard IEEE 802.21 QoS parameters to the access link technology-specific parameters. Table B.1 is informative and list of measurements defined in IEEE Std 802.11-2012 and the procedures that are described in its 10.24.9 should be used in computing the QoS performance metrics defined in this document. For IEEE 802.11, a collection of the QoS parameters can be on an individual station measurement basis, since this is a media using a distributed (symmetric) access technology.

Table B.1—QoS parameter mapping for IEEE 802.11

IEEE 802.21 link QoS parameters	Related IEEE 802.11 parameters	IEEE 802.11 IE name	Notes
Throughput	Not currently supported		Measurement is defined as the total number of octets transmitted/measurement duration.
Packet error rate	TransmittedFragmentCount MulticastTransmittedFrameCount FailedCount ReceivedFragmentCount (see NOTE) MulticastReceivedFrameCount FCSErrorCount (see NOTE) TransmittedFrameCount RetryCount MultipleRetryCount FrameDuplicateCount RTSSuccessCount RTSFailureCount ACKFailureCount	STA Statistics Report	
Supported number of COS	4 for IEEE 802.11, 8 for HCCA, 1 for non-IEEE 802.11 systems		
CoS minimum packet transfer delay	Transmit Delay Histogram (see NOTE)	Transmit Stream/Category Measurement Report	Trigger (option) (only to specific STA)
CoS average packet transfer delay	Average Transmit Delay	Transmit Stream/Category Measurement Report	Trigger (option) (only to specific STA)
CoS maximum packet transfer delay	Transmit Delay Histogram (see NOTE)	Transmit Stream/Category Measurement Report	Trigger (option) (only to specific STA)
CoS packet delay jitter	Transmit Delay Histogram (see NOTE) Average Transmit Delay (see NOTE)	Transmit Stream/Category Measurement Report	Trigger (option) (only to specific STA)
CoS packet loss rate	QoSTransmittedFragmentCount (see NOTE) QoSFailedCount QoSRetryCount QoSMultipleRetryCount QoSFrameDuplicateCount QoSRTSSuccessCount QoSRTSFailureCount QoSACKFailureCount (see NOTE) QoSReceivedFragmentCount QoSTransmittedFrameCount QoSDiscardedFrameCount QoSMPDUsReceivedCount QoSRetriesReceivedCount	STA Statistics Report	
	Transmitted MSDU Count (see NOTE) MSDU Discarded Count MSDU Failed Count (see NOTE) MSDU Multiple Retry Count QoS CF-polls Lost Count	Transmit Stream/Category Measurement Report	Trigger (option) (only to specific STA)
NOTE—This parameter is most likely to be used to directly derive IEEE 802.21 LinkQoSParameters. See B.3 for example derivations.			

Table B.2 and Table B.3 show example mappings for IEEE 802.21 QoS link parameters and other link specific parameters for IEEE 802.16, 3GPP, and 3GPP2. For these technologies control is usually by means of a base station, not an individual station, since the media is controlled using asymmetric access.

Table B.2—QoS parameter mapping for IEEE 802.16 and 3GPP2

IEEE 802.21 link QoS parameters	IEEE 802.16	3GPP2
Throughput	Maximum Sustained Traffic Rate	Peak_Rate
Packet loss rate		Max_IP_Packet_Loss_Rate
Packet error rate	Packet Error Rate	
CoS minimum packet transfer delay		
CoS average packet transfer delay		
CoS maximum packet transfer delay	Maximum Latency	Max_Latency
CoS packet delay jitter	Tolerated Jitter	Delay_Var_Sensitive

Table B.3—QoS parameter mapping for 3GPP

<table>
<tr><th>IEEE 802.21 link QoS parameters</th><th colspan="4">Related 3GPP parameters</th></tr>
<tr><td rowspan="2">Supported number of CoS</td><td colspan="4">4</td></tr>
<tr><td>Conversational</td><td>Streaming</td><td>Interactive</td><td>Background</td></tr>
<tr><td rowspan="4">Throughput</td><td colspan="4">Peak throughput</td></tr>
<tr><td colspan="4">Mean throughput</td></tr>
<tr><td colspan="4">Maximum bit rate for uplink/downlink</td></tr>
<tr><td colspan="2">Guaranteed bit rate for uplink/downlink</td><td></td><td></td></tr>
<tr><td rowspan="2">Link packet error rate</td><td colspan="4">SDU Error Ratio</td></tr>
<tr><td colspan="4">Residual Bit Error Rate</td></tr>
<tr><td>CoS minimum packet transfer delay</td><td colspan="2">Transfer delay</td><td></td><td></td></tr>
<tr><td>CoS average packet transfer delay</td><td colspan="2">Transfer delay</td><td></td><td></td></tr>
<tr><td>CoS maximum packet transfer delay</td><td colspan="2">Maximum Transfer delay</td><td></td><td></td></tr>
<tr><td>CoS packet transfer delay jitter</td><td></td><td>Delay variation</td><td></td><td></td></tr>
<tr><td rowspan="2">CoS packet loss rate</td><td colspan="4">Residual Bit Error Rate</td></tr>
<tr><td colspan="4">SDU Error Ratio</td></tr>
</table>

B.3 Deriving generic IEEE 802.21 QoS parameters

B.3.1 General

The following subclauses describe how to derive generic QoS parameters from IEEE Std 802.11-2012 link measurement parameters. This derivation relies on incremental values of counters as specified in IEEE Std 802.11-2012.

Note that the parameters are unicast parameters that are unrelated to multicast traffic.

B.3.2 Packet loss rate

To calculate the packet loss rate (PLR), one uses Equation (B.1).

$$\mathrm{PLR} = \frac{\text{the number of lost packets}}{\text{the number of transmitted packets (successful + failed)}} \tag{B.1}$$

According to IEEE Std 802.11-2012, a packet is a MAC user data packet or MAC service data unit (MSDU).

The PLR_{MSDU} can be derived from the Transmit Stream/Category Measurement Report using Equation (B.2).

$$PLR_{MSDU} = \frac{\text{failed MSDUs}}{\text{Transmitted MSDUs} + \text{Failed MSDUs}} \quad \text{(B.2)}$$

$$= \frac{\text{MSDU Failed Count}}{\text{Transmitted MSDUCount} + \text{MSDU Failed Count}}$$

B.3.3 Packet error rate

The packet error rate (PER) can be calculated using Equation (B.3).

$$PER = \frac{\text{the number of packets that are received with errors}}{\text{the number of packets in a population of interest}} \quad \text{(B.3)}$$

Unlike for PLR, this parameter is only defined for the IEEE 802.11 MPDU. The PER can be derived from the STA Statistics Report information element using Equation (B.4).

$$PER = \frac{\text{FCSERRORCount}}{\text{ReceivedFragmentCount} + \text{FCSErrorCount}} \quad \text{(B.4)}$$

B.3.4 Average transfer delay

In IEEE Std 802.11-2012, the transmit delay (MAC service data unit [MSDU] delay) is defined as follows:

Transmit delay (MSDU delay): The delay shall be measured from the time the MSDU is passed to the MAC sublayer until the point at which the entire MSDU has been successfully transmitted including receipt of the final ACK.

If the average MSDU transmit delay is used for the IEEE 802.21 average transfer delay, it can be derived from the Transmit Stream/Category Measurement Report.

ATD_{MSDU} = Average MSDU Transmit Delay

= Average Transmit Delay

B.3.5 Packet transfer delay jitter

Using the IEEE 802.21 definition of "the standard deviation of the delay over a population of interest," the IEEE 802.11 MAC sublayer provides the Transmit Stream/Category Measurement Report and measurement parameters to calculate the standard deviation of delay.

— QoS Metric information element includes:

a) Transmit Delay Histogram
b) Average Transmit Delay parameters

Variance calculation using discrete density function is given as

$$VAR(X) = \sum_{i=1}^{N} P_i (x_i - \bar{x})^2$$

Therefore, the packet transfer delay jitter for MSDU level is

Packet Transfer Delay Jitter = MSDU Packet Transmit Delay Jitter

$$= \sqrt{\sum_{i=1}^{N} P_i (x_i - \text{AverageTransmitDelay})^2}$$

where

N is the number of bins of Transmit Delay Histogram
P_i is the value (measured percentile) of i^{th} bin of Transmit Delay Histogram
x_i is the mean value of the delay range of i^{th} bin

Annex C

(normative)

Mapping media independent service (MIS) messages to reference points

Table C.1 maps the MIS messages to the MIS communication model reference points.

Table C.1—Mapping MIS messages to reference points

MIS message name	Reference point
MIS_Capability_Discover	RP1, RP2, RP3, RP4, RP5
MIS_Event_Subscribe	RP1, RP3
MIS_Event_Unsubscribe	RP1, RP3
MIS_Register	RP1, RP3, RP5
MIS_DeRegister	RP1, RP3, RP5
MIS_Link_Detected	RP1, RP3
MIS_Link_Up	RP1, RP3, RP5
MIS_Link_Down	RP1, RP3, RP5
MIS_Link_Parameters_Report	RP1, RP3, RP5
MIS_Link_Going_Down	RP1, RP3, RP2
MIS_Link_Get_Parameters	RP1, RP3, RP2
MIS_Link_Configure_Thresholds	RP1, RP3
MIS_Link_Actions	RP1, RP3
MIS_Get_Information	RP1, RP2, RP3, RP4, RP5
MIS_Push_Information	RP1, RP2, RP3, RP4, RP5
MIS_Auth	RP1, RP3
MIS_Termination_Auth	RP1, RP3
MIS_Push_Key	RP1, RP3
MIS_Configuration_Update	RP1, RP3, RP4, RP5
MIS_Pull_Group_Manipulate	RP1, RP3, RP4, RP5
MIS_Push_Group_Manipulate	RP1, RP3, RP4, RP5
MIS_Pull_Certificate	RP1, RP3, RP4, RP5
MIS_Push_Certificate	RP1, RP3, RP4, RP5
MIS_Revoke_Certificate	RP1, RP3, RP4, RP5

Annex D

(normative)

Media-specific mapping for service access points (SAPs)

The media independent services framework (MISF) aggregates disparate interfaces with respective media dependent lower layer instances (media dependent service access points) into a single interface with the MIS users (the MIS SAP), reducing the inter-media differences to the extent possible.

The MISF features media dependent interfaces with IEEE 802 link-layer technologies (IEEE 802.2, IEEE 802.3, IEEE 802.11, and IEEE 802.16) and cellular technologies (3GPP and 3GPP2). The MISF for the most part uses existing primitives and functionality provided by different access technology standards. Amendments to existing standards are recommended only when deemed necessary to fulfill the MISF capabilities.

The following subclauses list general amendments recommended to different underlying access technology standards due to the enhanced heterogeneous services (e.g., handover) capability provided by MISF.

D.1 MIS_LINK_SAP mapping to specific technologies

Table D.1—MIS_Link_SAP/IEEE 802.16 primitives mapping

MIS_LINK_SAP primitive		IEEE Std 802.16 C_SAP	IEEE Std 802.16 M_SAP
Link_Detected		C-HO-RSP (HO-Scan)	Not applicable (N/A)
Link_Up		C-NEM-RSP (Registration)	N/A
Link_Down		N/A	C-NEM-RSP (Deregistration)
Link_Parameters_Report		C-HO-IND (HO-Scan) C-HO-RSP (HO-Scan) C-RRM-RSP C-SFM-RSP	N/A
Link_Going_Down		N/A	N/A
Link_PDU_Transmit_Status		N/A	N/A
Link_Capability_Discover		N/A	N/A
Link_Event_Subscribe		N/A	N/A
Link_Event_Unsubscribe		N/A	N/A
Link_Get_Parameters		C-SFM-REQ/RSP C-HO-REQ/RSP/IND (HO-Scan) C-RRM-REQ/RSP	N/A
Link_Configure_Thresholds		C-HO-REQ/RSP (HO-Scan)	N/A
Link_Action	LINK_DISCONNECT	C-NEM-REQ/RSP (Deregistration)	N/A
	LINK_LOW_POWER	C-IMM-REQ/RSP (Idle_Mobile_Initiation)	
	LINK_POWER_DOWN	N/A	M-SSM-REQ/RSP (Power down)
	LINK_POWER_UP	N/A	M-SSM-REQ/RSP (Power on)

Table D.2—MIS_LINK_SAP/IEEE 802.11/IEEE 802.3/IEEE 802.1Q primitives mapping

Primitives	IEEE Std 802.11-2012	IEEE Std 802.3-2012	IEEE Std 802.1Q-2014
Link_Detected	MSGCF-ESS-Link-Detected[a]	N/A	N/A
Link_Up	MSGCF-ESS-Link-Up[a]	Link fault	dot 1agCfgFaultAlarm[b]
Link_Down	MSGCF-ESS-Link-Down[a]	Link fault	dot 1agCfgFaultAlarm[a]
Link_Parameters_Report	MLME-MEASURE.confirm MLME-MREPORT.indication[c] MSGCF-ESS-Link-Thresholdreport[a]	N/A	N/A
Link_Going_Down	MSGCF-ESS-Link-Going-Down[a]	Dying Gasp	N/A
Link_PDU_Transmit_Status	MA-UNIDATA-STATUS.indication	N/A	N/A
Link_Capability_Discover	N/A	N/A	N/A
Link_Event_Subscribe	N/A	N/A	N/A
Link_Event_Unsubscribe	N/A	N/A	N/A
Link_Get_Parameters	MSGCF-Get-ESS-Link-Parameters[a]	N/A	N/A
Link_Configure_Thresholds	MLME-MEASURE.request MLME-MREQUEST.request[d] MSGCF-Set-ESS-Link-Parameters[a]	N/A	N/A
Link_Action	MSGCF-ESS-Link-Command[a]	N/A	N/A

[a] See IEEE Std 802.11-2012.

[b] The alarms (cross-connection, link failure, MACstatusDefect, and RDIdefect) are enabled and no other higher priority event has occurred.

[c] IEEE 802.11 MLME-MEASURE.confirm and MLME-MREPORT.indication can be used. If MLME-MEASURE.request or MLME-MREQUEST.request includes Beacon Request IE or QoS Metric IE, then MLME-MEASURE.confirm or MLME-MREPORT.indication is delivered to the MISF when one of the reporting conditions (thresholds) is satisfied. Link_Parameter_Report.indication can be also generated at a predefined regular interval determined by a user configurable time. This is also performed by MLME-MEASURE.request and MLME-MEASURE.confirm (local) or MLME-MREQUEST.request and MLME-MREPORT.indication (remote) with measurement duration setting.

[d] It is used to configure threshold values for Link_Parameters_Report. Thresholds are used for triggering reports. IEEE 802.11 primitives, MLME-MEASURE.request(local) and MLME-MREQUEST.request(remote), can be used for that purpose. Only Beacon Request IE and QoS Metric IE can be used for setting thresholds and triggering reports. MLME-MEASURE primitive does not support confirmation to confirm the threshold setting results. It means that MLME-MEASURE primitive does not have the corresponding primitive to Link_Configure_Thresholds.confirm. MLME-MEASURE.confirm is used to deliver the measurement results not to confirm the threshold setting.

Table D.3—MIS_LINK_SAP/3GPP/3GPP2 primitives mapping

Primitives	3GPP	3GPP2
Link_Detected	N/A	N/A
Link_Up	SMSM-ACTIVE RABMSM-ACTIVATE	L2.Condition.Notification LCP-Link-Open LCP-Link-Up IPCP-Link-Open
Link_Down	SMSM-DEACTIVEATE SMSM-STATUS RABMSM-DEACTIVATE RABMSM-STATUS RABMAS-RAB-RELEASE	LCP-Carrier-Failure LCP-Link-Quality-Failure LCP-Timeout IPCP-Link-Closed IPCP-Config-Failure IPCP-Timeout
Link_Parameters_Report	SMSM-MODIFY RABMSM-MODIFY	N/A
Link_Going_Down	N/A	LCP-Closing
Link_PDU_Transmit_Status	N/A	N/A
Link_Capability_Discover	N/A	N/A
Link_Event_Subscribe	N/A	N/A
Link_Event_Unsubscribe	N/A	N/A
Link_Get_Parameters	N/A	N/A
Link_Configure_Thresholds	SMREG-PDP-MODIFY	L2.Supervision.Request
Link_Action	N/A	N/A

D.2 Mapping from MIS_LINK_SAP to media-specific SAPs

D.2.1 IEEE Std 802.3

Logical link control service access point (LSAP), defined in the ISO/IEC 8802-2:1998, provides the interface between the MISF and the LLC (Logical Link Control) sublayer in IEEE 802.3 network. This SAP is used for local media independent service (MIS) exchanges between the MISF and the lower layers of the IEEE 802.3 interface (as the IEEE 802.3 instantiation of the MIS_LINK_SAP) and for the L2 transport of MIS messages across IEEE 802.3 access links.

D.2.2 IEEE Std 802.11

The MISF uses MSGCF_SAP for interfacing with the link layer of IEEE 802.11 networks. The MIS_LINK_SAP defines additional primitives that map to MSGCF_SAP. These primitives are recommended as enhancements to IEEE 802.11 link-layer SAPs. MSGCF_SAP is defined by IEEE Std 802.11-2012 and it includes, but is not limited to primitives related to the following:

— System configuration
— Link state change notifications/triggers
— MIS frame transport through control or management frames

LSAP, defined in the ISO/IEC 8802-2:1998, provides the interface between the MISF and the LLC sublayer in IEEE 802.11. This SAP is used for the L2 transport of MIS messages across IEEE 802.11 access links. The MIS messages are carried in IEEE 802.11 data frames.

Table D.2 lists this mapping.

D.2.3 IEEE Std 802.16

The MISF uses C_SAP and M_SAP for interfacing with the Control and Management planes of the IEEE 802.16 network.

C_SAP is defined by IEEE Std 802.16-2012 and it includes primitives related to the following:

- Handovers (e.g., notification of handover request from mobile station [MS])
- Idle mode mobility management (e.g., mobile entering idle mode)
- Subscriber and session management (e.g., mobile requesting session setup)
- Radio-resource management
- Authentication, authorization, and accounting (AAA) server signaling (e.g., extensible authentication protocol [EAP] payloads)
- Media independent function services

M_SAP is defined by IEEE Std 802.16-2012 and it includes primitives related to the following:

- System configuration
- Monitoring statistics
- Notifications triggers
- Multi-mode interface management

CS_SAP, defined in the IEEE Std 802.16-2012, provides the interface between the MISF and the service-specific Convergence Sublayer in IEEE 802.16 networks. This SAP is used for the L2 transport of MIS messages through data frames across IEEE 802.16 access links.

Table D.1 list this mapping.

D.2.4 3GPP and 3GPP2

This SAP defines MIS_3GLINK_SAP interface between the MISF and the different protocol elements of the 3G system.

3GPP and 3GPP2 service primitives for GERAN, universal mobile telecommunications system (UMTS), Long-Term Evolution (LTE™[19]), cdma2000®[20], cdma2000-HRPD and UMB are used to access MIS services. This is done by establishing a relationship between the 3GPP/3GPP2 primitives and MIS primitives.

Table D.3 lists this mapping. Note that a 3GPP primitive group can be mapped to more than one MIS primitive, as shown in Table D.3.

[19] LTE is a trademark of The European Telecommunications Standards Institute (ETSI).
[20] cdma2000 is a registered trademark of the Telecommunications Industry Association (TIA-USA).

Annex E

(normative)

Data type definitions

E.1 General

This annex defines data types used in the IEEE 802.21 standard. Any variable-length data type in this specification contains information needed for determining the end of data.

E.2 Basic data types

The data types defined in this subclause are used as the basis for defining any other data types. All basic data types are for general purpose. The "binary encoding rule" column in Table E.1 describes the encoding rules used when the data types are carried in MIS protocol messages.

Table E.1—Basic data types

Data type name	Definition	Binary encoding rule
BITMAP(size)	A bitmap of the specified size. Usually used to represent a list of IDs. Range: Each bit has a value of '0' or '1'.	A BITMAP(N), where N shall be a multiple of 8, is made up of an N/8 octet values and encoded in network byte order.
CHOICE(DATATYPE1, DATATYPE2[,...])	A data type that consists of only one of the data types listed: DATATYPE1,DATATYPE2[,...].	A one-octet Selector field, followed by a variable length Value field. The Selector value determines the data type. If Selector==i, (i + 1)-th data type in the list of data types DATATYPE1,DATATYPE2[,...] is selected. The Selector value is encoded as UNSIGNED_INT(1). The Value field is encoded using the encoding rule for the selected data type.
INFO_ELEMENT	A binary encoded structure for Information Elements.	See 6.5.6.
INTEGER(size)	A signed integer of the specified size in number of octets. Range: Each octet has a value of 0x00 to 0xff.	Each octet of an INTEGER(N) value [N=1,2,...] is encoded in network-byte order into an N-octet field. The most significant bit of the first octet is the sign bit. If the sign bit is set, it indicates a negative integer. Otherwise, it indicates a non-negative integer. A negative integer is encoded as 2s complement.
LIST(DATATYPE)	A list of values of DATATYPE.	See E.2 for details.
NULL	A data type with empty data.	No octet is encoded for this data type. This data type is used to define an optional data type.
OCTET(size)	An array of octets. The size specifies the length.	The octets are encoded in network byte order.
SEQUENCE(DATATYPE1, DATATYPE2[,...])	A data type that consists of two or more data types.	DATATYPE1,DATATYPE2[,…] are encoded in the order of appearance. Each data type is encoded using the encoding rule for the data type.
UNSIGNED_INT(size)	An unsigned integer of the specified size in number of octets. Range: Each octet has a value of 0x00 to 0xff.	Each octet of an UNSIGNED_INT(N) value [N=1,2,...] is encoded in network-byte order into an N-octet field.

The encoding rule for LIST(DATATYPE) is a variable length *Length* field followed by a variable length *Value* field. The *Length* field shall be interpreted as follows:

Case 1: If the number of list elements in the *Value* field is less than 128, the size of the *Length* field is always one octet and the MSB of the octet is set to the value '0'. The values of the other seven bits of this octet indicate the actual number of list elements in the *Value* field.

Case 2: If the number of list elements in the *Value* field is exactly 128, the size of the *Length* field is one octet. The MSB of the *Length* octet is set to the value '1' and the other seven bits of this octet are all set to the value '0'.

Case 3: If the number of list elements in the *Value* field is greater than 128, then the *Length* field is always greater than one octet. The MSB of the first octet of the *Length* field is set to the value '1' and the remaining seven bits of the first octet indicate the number of octets that are appended further. The number represented by the second and subsequent octets of the *Length* field, when added to 128, indicates the total number of list elements in the *Value* field.

For example, an attribute of type LIST(LINK_ID) with two elements is encoded as shown in Figure E.1 (LINK_ID is defined in E.3.4):

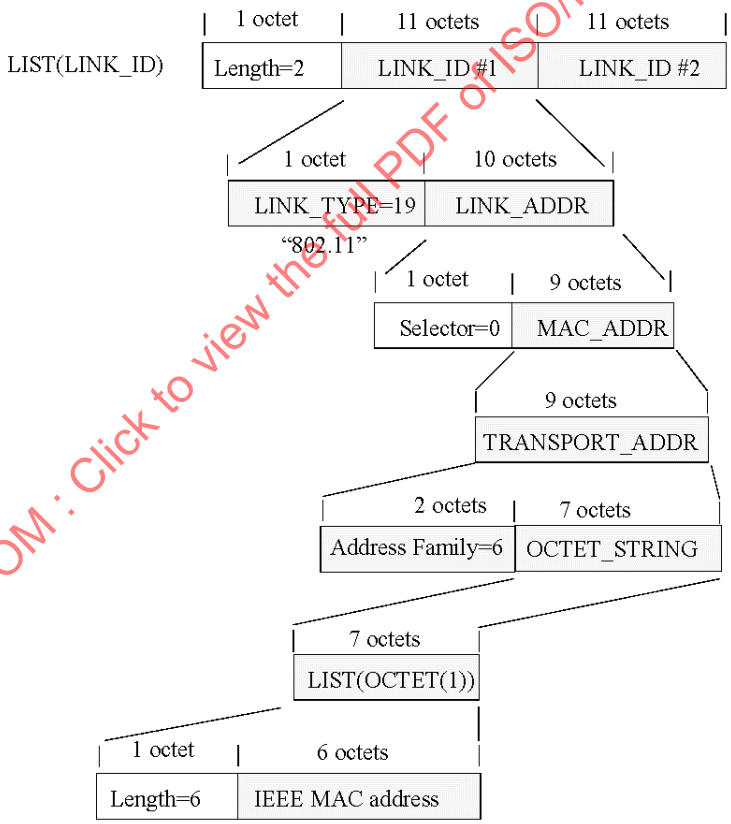


Figure E.1—Encoding example of a LIST with two LINK_ID elements