IEEE P802.21/D11.0

Draft Standard for Local and Metropolitan Area Networks: Media Independent Handover Services

Sponsor

LAN MAN Standards Committee

of the

IEEE Computer Society

Abstract: This standard specifies IEEE 802 media access-independent mechanisms that optimize handovers between heterogeneous IEEE 802 systems and between IEEE 802 systems and cellular systems.

Keywords: media independent handover

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

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

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

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

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

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

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

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

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

Introduction

This introduction is not part of IEEE P802.21/D11.0 Draft Standard for Local and Metropolitan Area Networks: Media Independent Handover Services.

This standard defines extensible media access independent mechanisms that enable the optimization of handovers between heterogeneous IEEE 802 systems and may facilitate handovers between IEEE 802 systems and cellular systems.

Participants

At the time this supplement to the standard was submitted to Sponsor Ballot, the working group had the following officers.

Vivek Gupta, *Chair*
Michael Williams, Vice Chair
Xiaoyu Liu, Secretary
Qiaobing Xie, Technical Editor

The following individuals served as the working group Assistant Editors:

Yuu-Heng Alice Cheng
David Cypher
Junghoon Jee
Marc Meylemans

Proposals for Media Independent Handover solutions were made by the following individuals (co-authors of the accepted proposal are marked with an asterisk)

Farooq Anjum*
Stefan Berg*
Yogesh Bhatt*
Alistair Buttar*
Alan Carlton*
Subir Das*
Ashutosh Dutta*
Les Eastwood*
Natraj Ekambaram*
Stefano Faccin*
Victor Fajardo*
Peretz Feder*
Andrea Francini*
Yuri Goldstein*
Prasad Govindarajan*
Wolfgang Groeting*
Vivek Gupta*
Youn-Hee Han*
Fang Hao*
Eleanor Hepworth*
Michael Hoghooghi*
Cheng Hong*
Prakash Iyer*
David Johnston*
Maged Kaki*
Ronny Kim*
Benjamin Koh*
Kalyan Koora*
Yongwon Kwak*
Jeff Keating*
Hong-Yon Lach*
Peter Larsson*
Jin Lee*
Sungjin Lee*
Michael Lee*
Xiaoyu Liu*
Guang Lu*
Sunil Madhani*
Mani Mahalingam*
Stephen McCann*
Ray Miller*
Padraig Moran*
Nat Natarajan*
Eric Njedjou*
Yoshihiro Ohba*

Ulises Olvera-Hernandez*
Chihyun Park*
Soohong Daniel Park*
Mathieu Peresse*
Daniel Orozco-Perez*
Akbar Rahman*
Ajay Rajkumar*
Sampath Rangarajan*
Marian Rudolf*
Reijo Salminen*
Sameer Sharma*
Ajoy Singh*
Kenichi Taniuchi*
Yousif Targali*
Michael Williams*
Qiaobing Xie*
Raziq Yaqub*
Tao Zhang*
Yao Zhonghui*
Juan Carlos Zuniga*

Others who made presentations to the Media Independent Handover Working Group, and contributions to this document, were:

Sanjeev Athayle
Gabor Bajko
Farooq Bari
Yuu-Heng Alice Cheng
David Cypher
David Famolari
Nada Golmie
Junxiang Guo
James Han
Kentaro Ishizu
Junghoon Jee
Hongseok Jeon
Farrokh Khatibi
Eunah Kim
Masahiro Kuroda
Jing Liu
Changmin Park
Yan Peng
Takashi Sakakura
Srinivas Sreemanthula
Lucian Suciu
Miriam Tauil
Albert Vidal
Jesse Walker
Bin Xia
Yi Zhou
Hongcheng Zhuang

[NOTE - Please send other names to the Editor. Need to update list and re-balance when list is done]

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

[To be supplied by IEEE]

Table of Content

Annexes

List of Figures

List of Tables

IEEE P802.21/D11.0

Draft Standard for Local and Metropolitan Area Networks: Media Independent Handover Services

1. Introduction

1.1 Scope

This standard defines extensible IEEE 802 media access independent mechanisms that enable the optimization of handover between heterogeneous IEEE 802 networks and facilitates handover between IEEE 802 networks and cellular networks.

1.2 Purpose

The purpose is to improve the user experience of mobile devices by facilitating handover between 802 networks whether or not they are of different media types, including both wired and wireless, where handover is not otherwise defined; and to make it possible for mobile devices to perform seamless handover where the network environment supports it. These mechanisms are also usable for handovers between 802 networks and non 802 networks.

1.3 Overview

This standard provides link layer intelligence and other related network information to upper layers to optimize handovers between heterogeneous networks. This includes media types specified by Third Generation (3G) Partnership Project (3GPP), 3G Partnership Project 2 (3GPP2), and both wired and wireless media in the IEEE 802 family of standards. In this standard, unless otherwise noted, *media* refers to 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:

— Intra-technology handover (except for handovers across extended service sets (ESSs) in case of IEEE 802.11);

— Handover policy;

— Security mechanisms;

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

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

The purpose of this standard is to enhance the experience of mobile users by facilitating handovers between heterogeneous networks. The standard addresses the support of handovers for both mobile and stationary users. For mobile users, handovers can occur when wireless link conditions change due to the users' movement. For the stationary user, handovers become imminent when the surrounding network environment changes, making one network more attractive than another.

This standard supports another important aspect of optimized handover - link adaptation. A user can choose 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 a handover if the higher rate is unavailable on the current link.

In all such cases service continuity should be maintained to the extent possible during handover. 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 will 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 networks. The network infrastructure is well-suited to store overall network information, such as neighborhood cell lists, location of mobile nodes, and higher layer service availability. Both the mobile node and the network make decisions about connectivity. In general, both the mobile node and the network points of attachment (such as base stations and access points) can be multi-modal (i.e., capable of supporting multiple radio standards and simultaneously supporting connections on more than one radio interface).

The overall network can include a mixture of cells of drastically different sizes, such as those from IEEE 802.15, IEEE 802.11, IEEE 802.16, 3GPP, and 3GPP2, with overlapping coverage. The handover process can be initiated by measurement reports and triggers supplied by the link layers on the mobile node. The measurement reports can include metrics such as signal quality, synchronization time differences, and transmission error rates. Specifically the standard consists of the following elements:

a) A framework that enables service continuity while a mobile node (MN) transitions between heterogeneous link-layer technologies. The framework relies on the presence of a mobility management protocol stack within the network elements that support the handover. The framework presents media independent handover (MIH) reference models for different link layer technologies.

b) A set of handover-enabling functions within the protocol stacks of the network elements and a new entity created therein called the MIH Function (MIHF).

c) A media independent handover Service Access Point (called the MIH_SAP) and associated primitives are defined to provide MIH Users with access to the services of the MIHF. The MIHF 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 MIH Users to control link properties that are relevant to handover and switch between links if required.

3) The Media Independent Information service provides the information about different networks and their services thus enabling more effective handover decision to be made across heterogeneous networks.

d) The definition of new link layer service access points (SAPs) and associated primitives for each link-layer technology. The new primitives help the MIHF collect link information and control link behavior during handovers. If applicable, the new SAPs are recommended as amendments to the standards for the respective link-layer technology.

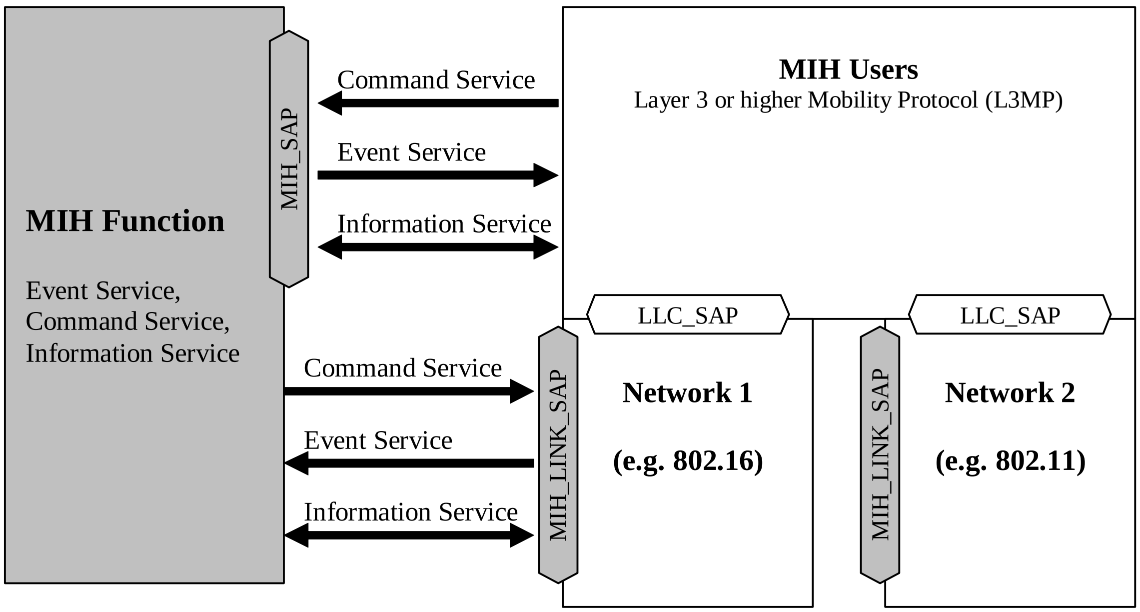


Figure 1 — MIH services and their initiation

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

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 MIHF is a logical entity, whose definition is independent of its deployment location on the MN or in the network;

c) The MIHF, 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 MIHF on the local network element obtains it through MIH message exchanges with a peer MIHF 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 MIHF on that entity obtains it locally through the service primitives of the SAPs that define the interface of the MIHF with the lower layers.

1.5 Media independence

The intent of this standard is to provide generic link layer intelligence independent of the specifics of mobile nodes or radio networks. As such this standard is intended to provide a generic interface between the link layer users in the mobility-management protocol stack and existing media-specific link layers, such as those specified by 3GPP, 3GPP2 and the IEEE 802 family of standards.

This standard defines SAPs and primitives that provide generic link layer intelligence. Individual media-specific technologies thereafter need to enhance their media-specific SAPs and primitives to satisfy the generic abstractions of this standard. Suitable amendments are required to existing link layer (medium access control (MAC)/ physical layer (PHY)) standards of different media-specific technologies such as IEEE Std 802.3, IEEE Std 802.11, IEEE Std 802.16, 3GPP, and 3GPP2 to satisfy the requirements of generic link layer intelligence identified by 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 (2007-09), Numbering, addressing and identification (Release 7).

3GPP TS 25.008, Digital cellular telecommunication system (Phase 2+); Radio subsystem link control.

3GPP TS 25.215 (2007-11), Physical layer - Measurements (FDD) (Release 7).

3GPP TS 25.401 (2007-09), UTRAN overall description (Release 7).

3GPP TS 25.413 (2007-09), UTRAN Iu interface RANAP signalling (Release 7).

3GPP2 C.S0004-D (2004-02), Signaling Link Access Control (LAC) Standard for cdma2000 Spread Spectrum Systems.

ANSI X3.159-1989: Programming Language C.

IEEE P802.11k™/D13.0, (2008-03), Draft 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; Amendment 1: Radio Resource Measurement of Wireless LANs.

IEEE P802.11u™/D2.0, (2008-01) (this version), Draft Amendment to 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: Amendment 7: Interworking with External Networks.

IEEE Std 802.16Rev2/D4.0, (2008-04), Information Technology- Telecommunications and information exchange between system-Local and metropolitan area networks-Specific Requirements-Part 16: Air Interface for Fixed Broadband Wireless Access Systems.

IEEE Std 802.11™-2007, 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™-2004 [ISO/IEC 8802-16: 2004], Information Technology- Telecommunications and information exchange between system-Local and metropolitan area networks-Specific Requirements-Part 16: Air Interface for Fixed Broadband Wireless Access Systems.

IEEE Std 802.16E™-2005 [ISO/IEC 8802-16: 2005], Information Technology- Telecommunications and information exchange between system-Local and metropolitan area networks-Specific Requirements-Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems (Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands).

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

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

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

IETF RFC 3344 (2002-08), IP Mobility Support for IPv4.

IETF RFC 3588 (2003-09), Diameter Base Protocol.

IETF RFC 3775 (2004-06), Mobility Support in IPv6.

IETF RFC 3825 (2004-07), Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information.

IETF RFC 4068 (2005-07), Fast Handovers for Mobile IPv6.

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

IETF RFC 4140 (2005-08), Hierarchical Mobile IPv6 Mobility Management (HMIPv6).

IETF RFC 4282 (2005-12), The Network Access Identifier.

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 4857 (2007-06), Mobile IPv4 Regional Registration.

IETF RFC 4881 (2007-06), Low-Latency Handoffs in Mobile IPv4.

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

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

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

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

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

W3C Recommendation, RDF/XML Syntax Specification.

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

W3C Recommendation, SPARQL Query Language for RDF.

3. Definitions

For the purpose of this standard, the following terms and definitions apply. The Authoritative Dictionary of IEEE Standards Terms should be referenced for terms not defined in this clause.

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

3.2 dual-radio operation: In this mode a dual radio device can receive and transmit simultaneously on both the radios. Since both radios can be active simultaneously in these types of devices, the target radio connects with the target network to prepare the target network for handover. The source radio maintains connection with the source network during the handover. *See also* **single-radio operation**.

3.3 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 can 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**.

3.4 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).

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

3.6 home subscriber network: Network managed by an operator with whom the subscriber has a business relationship (subscription). *See also* **visited network; serving network.**

3.7 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) to UMTS or wireless local area network (WLAN) to WLAN. *Syn:* **intra-technology handovers.**

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

3.9 intra-technology handovers: *See*: **horizontal handovers.**

3.10 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.

3.11 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.

3.12 link indication: Link state information provided by the link layer to higher layers.

3.13 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.

3.14 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), L1/LAPDm in the case of GSM CS, and L1/MAC/RLC in the case of general packet radio service (GPRS)/ Enhanced GPRS (EGPRS), respectively. The term “Lower Layers” also includes Logical Link Control Layers such as IEEE 802.2 Logical Link Control (LLC) or 3GPP Radio Link Control (RLC). The MIHF uses the services provided by these layers.

3.15 MIH discovery protocol: A protocol for discovering media independent handover (MIH) entities.

3.16 MIH network entity: Network Entity with MIHF capability.

3.17 MIH point of service (MIH PoS): Network-side MIHF instance that exchanges MIH messages with an MN-based MIHF. The same MIH Network Entity includes an MIH PoS for each MIH-enabled mobile node with which it exchanges MIH messages. A single MIH PoS can host more than one MIH service. The same MIH Network Entity can include multiple MIH Points of Service that can provide different combinations of MIH services to the respective mobile nodes based on subscription or roaming conditions. Note that for a network entity comprising multiple interfaces, the notion of MIH PoS is associated with the network entity itself and not with just one of its interfaces.

3.18 MIH node: An MIHF capable entity (mobile node or network).

3.19 MIH non-PoS: An MIH network entity that can directly exchange MIH messages with other MIH network entities but cannot *directly* exchange MIH messages with any MIH enabled mobile node.

3.20 MIH transport protocol: A protocol for transporting MIH protocol messages between a pair of MIH entities.

3.21 MIH Users: Entities that use the services provided by the MIHF. MIH Users use the MIH_SAP to interact with the MIHF.

3.22 media independent handover function (MIHF): A function that realizes MIH services.

3.23 MIHF pairing: The communication relationship that exists between different MIHF instances when they exchange MIH messages or MIH information.

3.24 MIHF transaction: A combination of an MIH Request message and MIH Response message, MIH Indication, or MIH Response message and any associated MIH Acknowledgement messages.

3.25 mobile-controlled handover: The mobile node has the primary control over the handover process.

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

3.27 mobile node (MN): Communication node that can change its Point of Attachment from one link to another.

3.28 mobile node association: The connectivity state where the mobile node is ready to exchange user data (like transmission control protocol (TCP) / user datagram protocol (UDP) packets) with the network point of attachment.

3.29 network detection: The process by which a mobile node collects information on networks in its locality, identifies the different points of attachment, and ascertains the validity of link-layer configuration.

3.30 network entity: A communication node inside the network.

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

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

3.33 network neighborhood: 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.

3.34 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**.

3.35 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.

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

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

3.38 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.

3.39 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.

3.40 serving network: A network that provides services to the user. The serving network can be a home subscriber network or a visited network. *See also* **visited network; home subscriber network.**

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

3.42 single-radio operation: In this mode a dual radio device can receive and transmit on only one radio at a time. This is usually the mode of operation when radio frequencies of the two radios are close to each other (e.g. in IMT 2000 bands). Since only one radio can be active at a time in these types of devices, the source radio uses the back-end connection of the source network with the target network to prepare the target network for handover while maintaining the client side connections. Once the target preparation is complete the device switches from source radio to target radio. Since all the target preparation has been completed a prior, the target radio quickly establishes connectivity with the target network and all the connections are then transferred from source network to target network. *See also* **dual-radio operation**.

3.43 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).

3.44 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).

3.45 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.

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

3.47 vertical handovers: A handover where the mobile node moves between point of attachments of different link types, such as from UMTS to WLAN. *Syn:* **inter-technology handovers**

3.48 visited network: A network managed by an operator other than the subscriber's home operator and in which the subscriber is receiving service. *See also* **home subscriber network; serving network.**

4. Abbreviations and acronyms

The following abbreviations and acronyms are used in this standard.

3G	3rd generation
3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AAA	authentication, authorization, and accounting
ACK	acknowledgement
AID	action identifier
AP	access point
AR	access router
BCE	binding cache entry
BS	base station
BTS	base transceiver station
CoA	care-of address
CoS	class of service
CS	convergence sublayer / command service
DHCP	dynamic host configuration protocol
ES	event service
ESS	extended service set
FA	foreign agent
GPRS	general packet radio service
GSM	global system for mobile communication
HESSID	homogenous extended service set ID
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	internet protocol
IS	information service
ITU	International Telecommunications Union
L1	layer 1 (PHY)
L2	layer 2 (MAC and/or LLC)
LAN	local area network
LbyR	location by reference
LCP	location configuration protocol
LLC	logical link control
LMA	local mobility anchor
LSAP	logical link control service access point
LTE	long term evolution
MAC	medium access control
MAG	mobile access gateway
MICS	media independent command services
MIES	media independent event services
MIH	media independent handover
MIHF	media independent handover function
MIIS	media independent information service
MIP	mobile IP
MLME	MAC layer management entity
MN	mobile node
MPLS	multi-protocol label switching
MS	mobile station
MSB	most significant bit
MSDU	medium access control (MAC) service data unit
MSGCF	MAC state generic convergence function

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
PKMv2	privacy and key management version two
PLME	physical layer management entity
PLMN	public land mobile network
PoA	point of attachment
PoS	point of service
PPP	point-to-point protocol
PSAP	public safety answering point
QoS	quality of service
RAT	radio access technology
RDF	resource description framework
RFC	request for comment
RLC	radio link control
RNC	radio network controller
RSNA	robust security network association
RSSI	received signal strength indication
SAP	service access point
SCTP	stream control transmission protocol
SDO	standards development organization
SDU	service data unit
SIB	system information block
SID	service identifier
SINR	signal over interference plus noise ratio
SIP	session initiation protocol
SLA	service level agreement
SM	session management
SME	station management entity
SNR	signal-to-noise ratio
SS	subscriber station
STA	station
TCP	transmission control protocol
TLV	type-length-value
UDP	user datagram protocol
UE	user equipment
UIR	unauthenticated information request
UMTS	universal mobile telecommunications system
URL	uniform resource locator
WEP	wired equivalent privacy
WLAN	wireless local area network
XML	extensible mark-up language

5. General architecture

5.1 Introduction

5.1.1 General

This standard supports different handover methods. Such methods are generally classified as “hard” or “soft”, depending on whether the handover procedure is “break-before-make” or “make-before-break” with respect to the data transport facilities that support the exchange of data packets between the MN and the network.

Handover decision making involves cooperative use of both MN and network infrastructure. Handover control, handover policies and other algorithms involved in handover decision making are generally handled by communication system elements that do not fall within the scope of this standard. However, it is beneficial to describe certain aspects of the overall handover procedure so that the role and purpose of the MIH services in the handover process are clear. The following subclauses give an overview of how the different factors that affect handovers are addressed within this standard.

5.1.2 Service continuity

Service continuity is defined as the continuation of the service during and after the handover while minimizing aspects such as data loss and duration of loss of connectivity during the handover without requiring any user intervention. The change of access network need not be noticeable to the end user. However irrespective of that, there should be no need for the user to re-establish the service. There can be a change in service quality as a consequence of the transition between different networks due to the varying capabilities and characteristics of the access networks. For example if the quality of service (QoS) supported by the new access network is unacceptable, higher layer entities can decide not to handover or terminate the current session after the handover based on applicable policies. This standard specifies essential elements that enable service continuity.

5.1.3 Application class

Various applications have different tolerance characteristics for delay and data loss. Application aware handover 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.

5.1.4 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 during handover.

There are two aspects of QoS to consider in the context of IEEE 802.21. Firstly, there is the QoS experienced by an application during a handover. Secondly, there is the QoS considered as part of a handover decision. This standard includes mechanisms that support both aspects of QoS towards enabling seamless mobility; however the MIHF alone cannot guarantee seamless mobility. Depending on the QoS requirements of the end-to-end application, seamless mobility 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.

The MIH QoS model defines parameters that are used to set the requirements and assess the performance of packet transfers between a source and its destinations. When used in threshold-setting commands (such as MIH_Link_Configure_Thresholds), these parameters describe the QoS requirements of the MIH User. On the other hand, when used in parameter-reporting events (such as MIH_Link_Parameters_Report) and parameter-extraction commands (such as MIH_Link_Get_Parameters), they characterize current network conditions. Therefore, depending on their usage these parameters can represent either static QoS requirements or dynamic network measurements.

5.1.5 Network discovery

This standard defines the information that helps in network discovery and specifies the means by which such information can be obtained and be made available to the MIH Users. The network information includes information about link type, link identifier, link availability, link quality, etc.

5.1.6 Network selection

Network selection is the process by which an MN or a network entity selects a network (possibly out of many available) to establish network-layer connectivity. The selection is based on various criteria such as required QoS, cost, user preferences, or the network operator's policies. This standard specifies means by which such information can be made available to the MIH Users to enable effective network selection.

5.1.7 Power management

This standard allows the MN to discover different types of wireless networks (e.g. 802.11, 802.16 and 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 discovery of potential handover candidates. Specific power management mechanisms deployed are dependent on individual link-layer technologies and the potential power management benefits from this standard only extends to the discovery of wireless networks.

5.1.8 Handover policy

The primary role of the MIHF is to facilitate handovers and provide intelligence to the network selector entity. The MIHF 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.

5.2 General design principles

5.2.1 MIHF design principles

This standard is based on the following general design principles.

a) MIHF is a logical entity that facilitates handover decision making. MIH Users make handover decisions based on inputs from the MIHF.

b) MIHF 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 MIHF communicates with the lower layers of the mobility-management protocol stack through technology-specific interfaces.

c) Higher layer mobility management protocols specify handover signaling mechanisms for vertical handovers. Additionally, different access network technologies have defined handover signaling mechanisms to facilitate horizontal handover. The definition of such handover signaling mechanisms is outside the scope of this standard except in the case of handovers across ESSs in 802.11.

The role of this standard is to serve as a handover facilitating service and to maximize the efficiency of such handovers by providing appropriate link layer intelligence and network information.

d) The standard provides support for remote events. Events are advisory in nature. 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 standard 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 International Telecommunications Union (ITU) - Telecommunication Standardization Sector (ITU-T) Recommendation Y.1540.

It is generally accepted that, based on the required accuracy of information transfer, applications can be grouped into a small number of behavioral sets (ITU-T recommendation Y.1540) called Class of Service (CoS). Support for differentiation via Classes of Service is pervasive in many of the IEEE 802 based standards (IEEE Std 802.11, IEEE Std 802.1q, IEEE Std 802.16, etc.).

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

5.3 MIHF service overview

5.3.1 General

This standard defines services that comprise the MIHF service; these services facilitate handovers between heterogeneous access links.

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 MIH 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 MIHF provides asynchronous and synchronous services through well-defined SAPs for link layers and MIH Users. In the case of a system with multiple network interfaces of arbitrary type, the MIH Users use the Event Service, Command Service and Information Service provided by MIHF to manage, determine, and control the state of the underlying interfaces.

These services provided by MIHF help the MIH Users in maintaining service continuity, service adaptation to varying quality of service, battery life conservation, network discovery, and link selection. In a system containing heterogeneous network interfaces of IEEE 802 types and cellular (3GPP, 3GPP2) types, the MIHF helps the MIH Users to implement effective procedures to couple services across heterogeneous network interfaces. MIH Users utilize services provided by the MIHF across different entities to query resources required for a handover operation between heterogeneous networks.

MIH Services in mobile nodes facilitate seamless handovers between heterogeneous networks. MIH Services are used by MIH Users such as a mobility management protocol (e.g., Mobile IP). Other mobility management protocols (in addition to Mobile IP) and even other MIH Users are not precluded from making use of MIH Services.

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 are 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 MIHF (MIH Events) or any lower layer (Link Events) within the protocol stack of an MN or network node, as shown in Figure 12.

5.3.2.3 Event destination

The destination of an event is the MIHF 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 MIHF and from MIHF to any upper layer. In case of remote events, messages propagate from the MIHF in one protocol stack to the MIHF in the peer protocol stack. One of the protocol stacks can be present in an MN while the other can be 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 handovers. For example, an indication that the link will cease to carry MAC service data units (SDUs) at some point in the near future is used by MIH 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 to handover between attachment points.

Events carry additional context data such as a layer 2 (MAC and/or LLC) (L2) identifier or L3 identifier. A Link_Up event can also carry 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. If an MIHF supports the command service, all MIH commands are mandatory in nature. When an MIHF receives a command, it is always expected to execute the command.

5.3.3.2 Command origination

Commands are invoked by MIH Users (MIH Commands), as well as by the MIHF itself (Link Commands), as shown in Figure 15.

5.3.3.3 Command destination

The destination of a command is the MIHF 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 MIH Users (e.g., policy engine) to the MIHF and then from MIHF to lower layers. In the case of remote commands, messages propagate from MIH Users via MIHF in one protocol stack to the MIHF in a peer protocol stack (with the use of the MIH Protocol). One of the protocol stacks can be present in an MN while the other can be 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 functions

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 can be 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).

This standard facilitates both mobile-initiated and network-initiated handovers. Handovers are initiated by changes in the wireless environment that leads to the selection of a network that supports a different access technology other than the serving network.

During network selection, the MN and the network need to exchange information about available candidate networks and select the best network. The network selection policy engine can select a different network than the current one, which can necessitate an inter-technology handover. Network selection and handover initiation are outside the scope of mobility management protocols such as mobile IP (MIP) and session initiation protocol (SIP). Once a new network has been selected and handover has been initiated, mobility management protocols handle packet routing aspects such as address update and transfer of packet delivery to the new network.

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 handovers and exchange information about available networks and negotiate the best available network under different conditions. Please refer to the flow diagrams in Annex L for more information. These commands do not affect packet routing aspects and can be used in conjunction with other mobility management protocols such as MIP and SIP to perform inter-technology handovers.

5.3.4 Media independent information service

The Media Independent Information Service (MIIS) provides a framework and corresponding mechanisms by which an MIHF entity can discover and obtain network information existing within a geographical area to facilitate the handovers.

Additionally or alternatively, the neighboring network information discovered and obtained by this framework and mechanisms can also be 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 for information transfer. This contrasts with the asynchronous push model of information transfer for the event service. The information can be present in some information server from where the MIHF 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 both a secure and a non-secure port. Information available through the non-secure port allows a network selection decision to be made before incurring the overhead of authentication and the establishment of a secure L2 connection with the network.

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

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 can also help in more effective handover decision making before the MN actually attaches to any particular network.

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 mobility management will collect information from different media types and assemble a consolidated view to facilitate its inter-media handover decision.

Some networks such as the cellular networks already have an existing means of detecting a list of neighborhood base stations within the vicinity of an area via the broadcast control channel. Some IEEE standards define similar means and support MNs in detecting a list of neighborhood 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 handover candidate information in a heterogeneous network environment by a given geographical location. However, the algorithm for deciding what information to provide is out of scope. In the larger view, the objective is to help the higher layer mobility protocol to acquire a global view of the heterogeneous networks to effect seamless handover across these networks.

5.4 Media independent handover reference framework

5.4.1 General

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

5.4.2 MIHF communication model

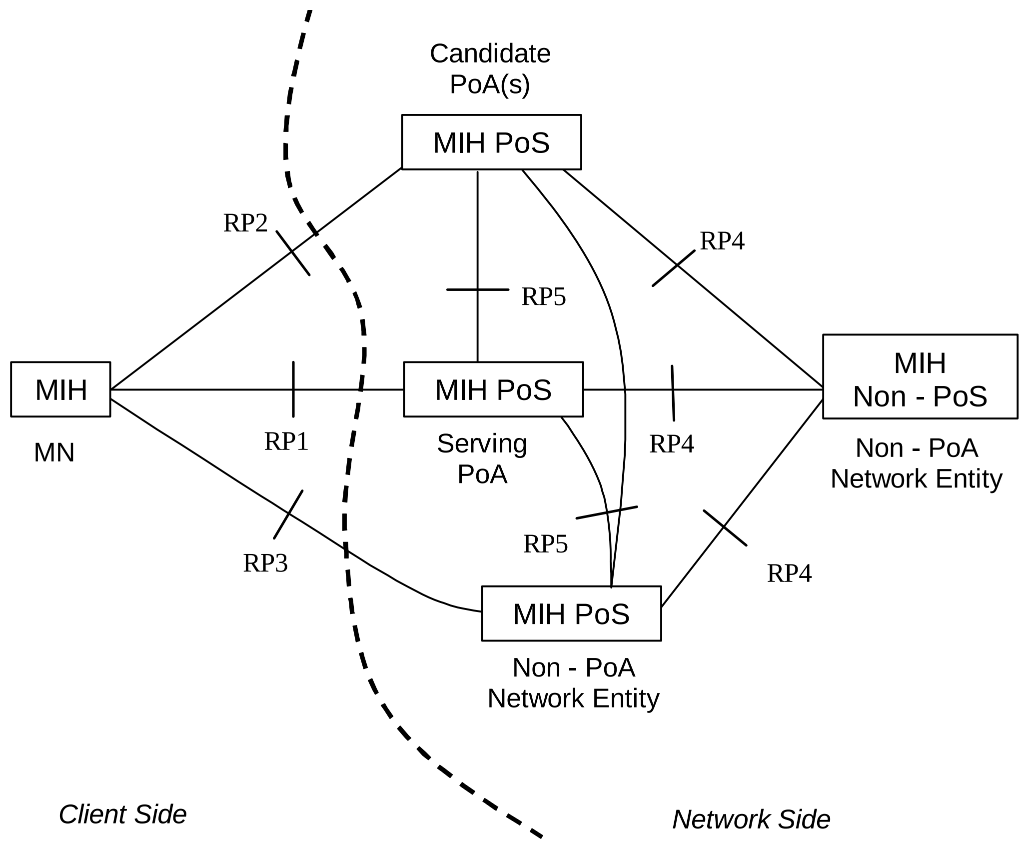


Figure 2 — MIHF communication model

MIH Functions communicate with each other for various purposes. The MN exchanges MIH information with its MIH Point of Service. The MIHF in any Network Entity becomes an MIH point of service (PoS) when it communicates directly with an MN-based MIHF. When an MIHF in a Network Entity does not have a direct connection to the MN, it does not act as an MIH PoS for that particular MN. However the same MIH Network Entity can still act as MIH PoS for a different MN.

A MN can have multiple L2 interfaces. However MIHF communication need not take place on all L2 interfaces of an MIH-capable MN. As an example, on an MIH-capable MN with three L2 interfaces, namely IEEE 802.11, IEEE 802.16, and IEEE 802.3, the IEEE 802.3 interface might be used only for system administration and maintenance operations, while the IEEE 802.11 and IEEE 802.16 interfaces might engage in the provisioning of MIHF services. The MN can use L2 transport for exchanging MIH information with an MIH PoS that resides in the same Network Entity as its Network PoA. The MN can use L3 transport for exchanging MIH information with an MIH 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 MIH network entities.

Figure 2 shows the MIHF communication model. The model shows MIHFs in different roles and the communication relationships amongst them. The communication relationship shown in Figure 2 applies only to MIHFs. 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 MIHF related information is possible between the two different MIHFs. Moreover, each communication relationship shown in the diagram encompasses different types of interfaces, different transport mechanisms used (e.g., L2, L3), and different MIHF service related content being passed (e.g., MIIS, MICS, or MIES).

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

a) MIHF on the MN
b) MIH PoS on the Network Entity that includes the serving PoA of the MN
c) MIH PoS on the Network Entity that includes a candidate PoA for the MN
d) MIH PoS on a Network Entity that does not include a PoA for the MN
e) MIH 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 MIHFs (see Table 1).

— **Reference point RP1**: Reference point RP1 refers to MIHF procedures between the MIHF on the MN and the MIH PoS on the Network Entity of its serving PoA. RP1 encompasses communication interfaces over both L2 and L3 and above. MIHF content passed over RP1 are related to MIIS, MIES, or MICS.
— **Reference point RP2**:Reference point RP2 refers to MIHF procedures between the MIHF on the MN and the MIH PoS on the Network Entity of a candidate PoA. RP2 encompasses communication interfaces over both L2 and L3 and above. MIHF content passed over RP2 are related to MIIS, MIES, or MICS.
— **Reference point RP3**: Reference point RP3 refers to MIHF procedures between the MIHF on the MN and the MIH 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). MIHF content passed over RP3 are related to MIIS, MIES, or MICS.
— **Reference point RP4**: Reference point RP4 refers to MIHF procedures between an MIH PoS in a Network Entity and an MIH non-PoS instance in another Network Entity. RP4 encompasses communication interfaces over L3 and above. MIHF content passed over RP4 are related to MIIS, MIES, or MICS.
— **Reference point RP5**: Reference point RP5 refers to MIHF procedures between two MIH PoS instances in different Network Entities. RP5 encompasses communication interfaces over L3 and above. MIHF content passed over RP5 are related to MIIS, MIES, or MICS.

Table 1—Summary of reference points

Reference point	Description
RP1	Between the MIHF on an MN and an MIH PoS on the Network Entity of the serving PoA.
RP2	Between the MIHF on an MN and an MIH PoS on the Network Entity of the candidate PoA.
RP3	Between the MIHF on an MN and an MIH PoS on a non-PoA network entity.
RP4	Between an MIHF PoS and an MIH non-PoS instance in different Network Entities.
RP5	Between two MIH PoS instances in different Network Entities.

All reference point definitions are within the scope of this standard. Table H-1 provides a mapping of various MIH messages to the reference points.

5.4.3 A deployment example for the MIH services

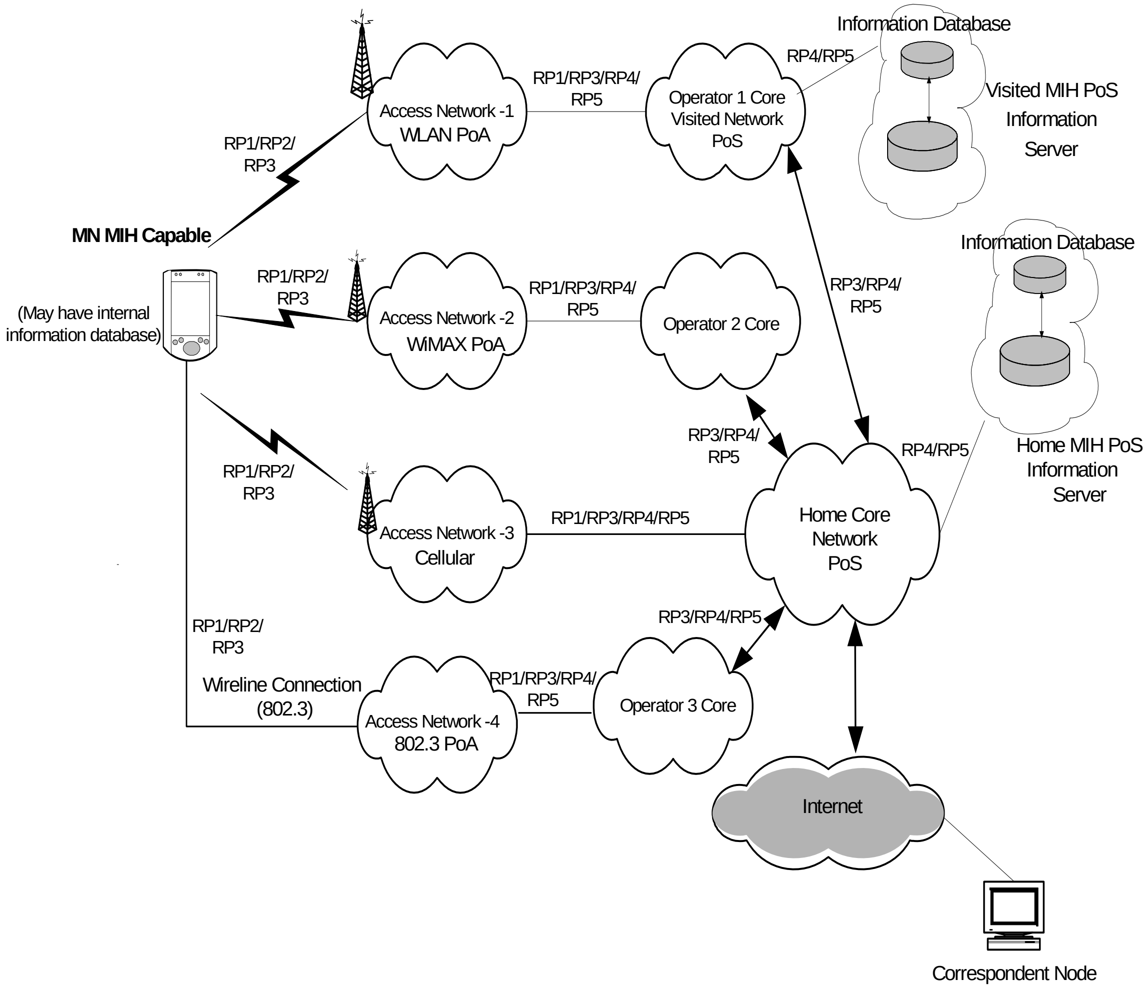


Figure 3 — Example of network model with MIH services

A network model including MIH services is shown in Figure 3 to better illustrate the MIH Reference Points. Moving from left to right, the model includes an MIH-capable mobile node (MN, far left) that supports multiple wired and wireless access technologies. The model assumes that the serving network either operates multiple link-layer technologies or allows its user to roam into other networks when a service level agreement (SLA) in support of inter-working has been established.

The model illustrates access networks that are connected in some loose, serial way to a given core network (i.e., Core Operator 1, 2, or 3). Also depicted is an access network that is more tightly coupled (Access Network-3). Not depicted in Figure 3, an access network can also connect to a core network via the Internet. Each Core Operator network (1, 2, or 3) might represent a service provider, corporate intranet provider, or just another part of the visited or home access. In this depicted model the provisioning provider is operating Access Network-3, which couples the terminal to the core (labeled Home Core Network) via RP1. At any given point in time, the subscriber's serving network can be the home subscriber network or a visited network.

The network providers offer MIH services in their access networks (Access Network-1 to 4) in order to facilitate heterogeneous handovers into their networks. Each access technology either advertises its MIH capability or responds to MIH service discovery. Each service provider for these access networks allows access to one or more MIH Points of Service (PoS) node(s). These PoS nodes provide some or all of the

MIH services as determined during the MIH capabilities discovery. The PoS location varies based on the operator deployment scenario and the technology-specific MIH architecture.

An MIH PoS resides next to, or is co-located with, the point of attachment (PoA) node in the access network (e.g., Access Network 1, 2, 4). Alternatively the PoS can reside deeper inside the access or core networks (e.g., Access Network 3). As shown in Figure 3, the MIH entity in the MN can communicate with MIH network entities using reference points RP1, RP2, or RP3 over any of the available access network. If the PoA in the serving access network has a co-located MIHF, the RP1 reference point terminates at the PoA that is also the PoS (MN to Access Network 1, 2, 4 of the model can all be RP1). In that case an RP3 reference point would be terminated at any non-PoA (illustrated by MN connectivity to Access Networks 1, 2, 4). MIH events originate at both sides of an active RP1 link. The MN is typically the first node to react to these events.

The interaction of visited and home subscriber networks could be either for control and management purposes or for data transport purposes. It is also possible that due to roaming or SLA agreements, the home subscriber network allows the MN to access the public Internet directly through a visited network. As illustrated, two MIH network entities communicate with each other via RP4 or RP5 reference points. The MIH capable PoA communicate with other MIH network entities via RP4 and RP5 reference points. The MIH capable MN have MIH communication with other PoA in the candidate access networks via RP2 reference point to obtain Information Services about the candidate network.

With regard to the MIH Information Service, visited providers can offer access to their information server located in an MIH PoS node (upper far right). The operator provides the MIIS to mobile nodes so they can obtain pertinent information including, but not limited to, new roaming lists, costs, provider identification information, provider services, priorities and any other information that would enable the selection and utilization of these services. As illustrated, it is possible for the MN to be pre-provisioned with MIIS data by its provider. It is also possible for the MN to obtain MIH Information Services from any access network of its service provider or from visited networks that maintain SLA agreements with the MN's service provider. MIIS can also be available from another overlapping or nearby visited network, using that network's MIIS point of service. The serving network utilizes RP4 and RP5 interfaces to access other MIH entities. As an example, in Figure 3 the home subscriber network accesses its own MIH information server or core operator 1 (visited network) MIH information server.

5.5 MIHF reference models for link-layer technologies

The MIHF provides asynchronous and synchronous services through well-defined Service Access Points for MIH Users. The following subclauses describe the reference models for various link-layer technologies with MIH functionality.

5.5.1 IEEE 802 architectural considerations

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

The MIH Protocol provides the capability for transferring MIH messages between peer MIH Function entities at L2 or at L3. These messages transfer information about different available networks and also provide network switching and handover capability across different networks. The MIH protocol encompasses IEEE 802 technologies such as 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 MIH Protocol has different scope and functionality than the Link layer Discovery Protocol (LLDP) as specified by the IEEE Std 802.1AB.

5.5.2 General MIHF reference model and SAPs

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

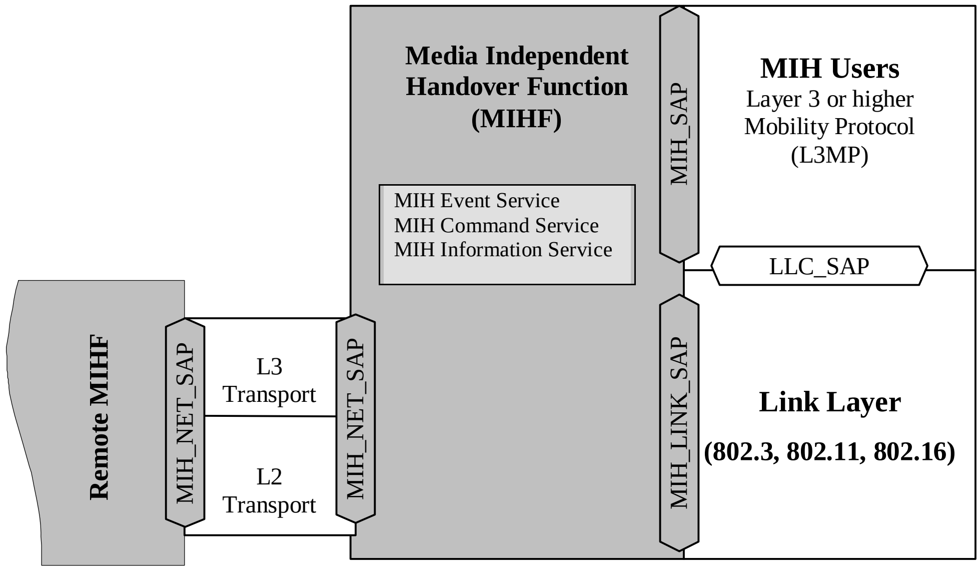


Figure 4 — General MIHF reference model and SAPs

The media agnostic General MIH Reference Model includes the following SAPs:

a) MIH_SAP: Media independent interface of MIHF with the upper layers of the protocol stack.

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

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

In the media-specific reference models, the media independent SAP (MIH_SAP) always maintains the same name and same set of primitives. The media dependent SAP (which is a technology specific instantiation of the MIH_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 MIH_LINK_SAP result in amendments to media-specific SAPs due to additional functionality being defined for interfacing with the MIHF. All communications of the MIHF with the lower layers of media-specific protocol stacks take place through media-specific instantiations of MIH_LINK_SAP.

The message exchanges between peer MIHF 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 MIHF 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 MIH capabilities at the PoA.

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

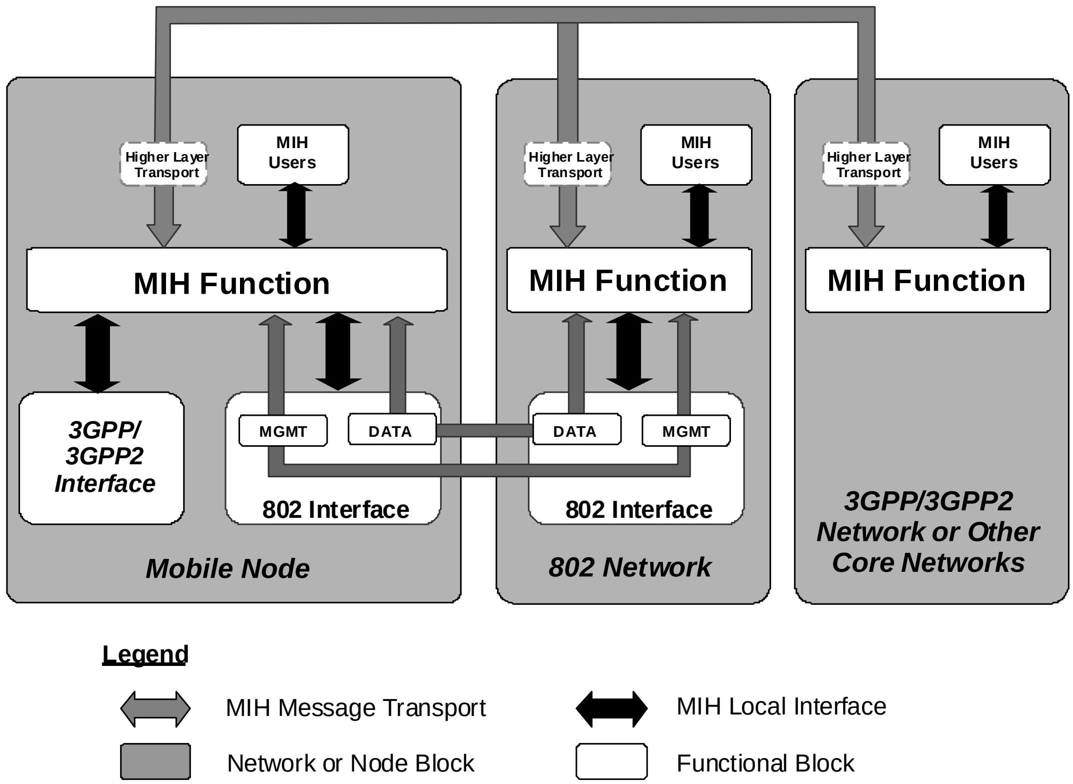


Figure 5 — Types of MIHF relationship

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

Figure 5 shows how an MIH-enabled MN communicates with an MIH-enabled network. The gray arrows show the MIH signaling over the network, whereas the black arrows show local interactions between the MIHF 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 the following subclauses.

When connected to an IEEE 802 network, an MN directly uses L2 for exchanging MIH signaling, as the peer MIHF 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 MIH signaling, for example in cases where the peer MIHF 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 MIH signaling.

5.5.3 MIHF reference model for IEEE 802.3

The MIHF reference model for IEEE 802.3 is illustrated in Figure 6. The transport of MIHF 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 to support any link services defined over the MIH_LINK_SAP in this specification.

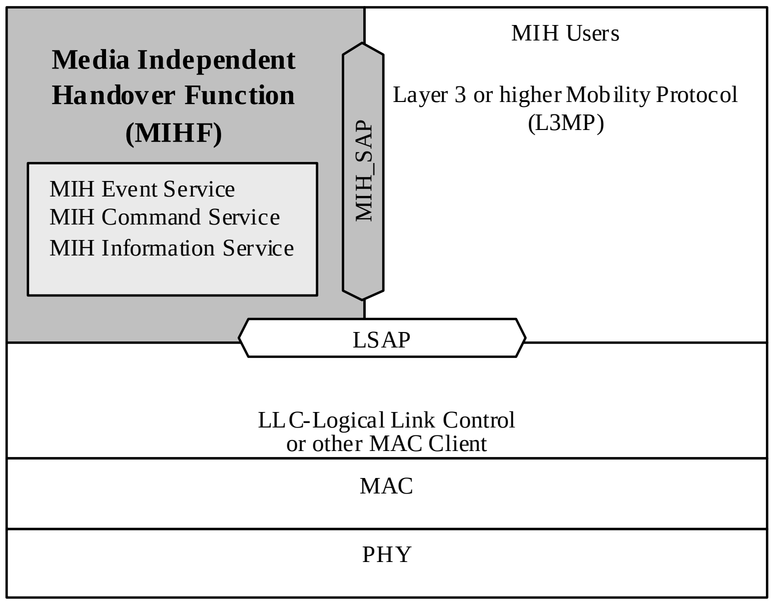


Figure 6 — MIH reference model for IEEE 802.3

5.5.4 MIHF reference model for IEEE 802.11

Figure 7 shows the MIHF reference model for IEEE 802.11. The payload of MIHF 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 layer management entity (MLME) service access point (SAP) (MLME_SAP).

It should be noted that sending MIHF 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 MIH Information Service and MIH Capability Discovery messages can be transported over the MLME_SAP.

The MIH_SAP specifies the interface of the MIHF with MIH Users.

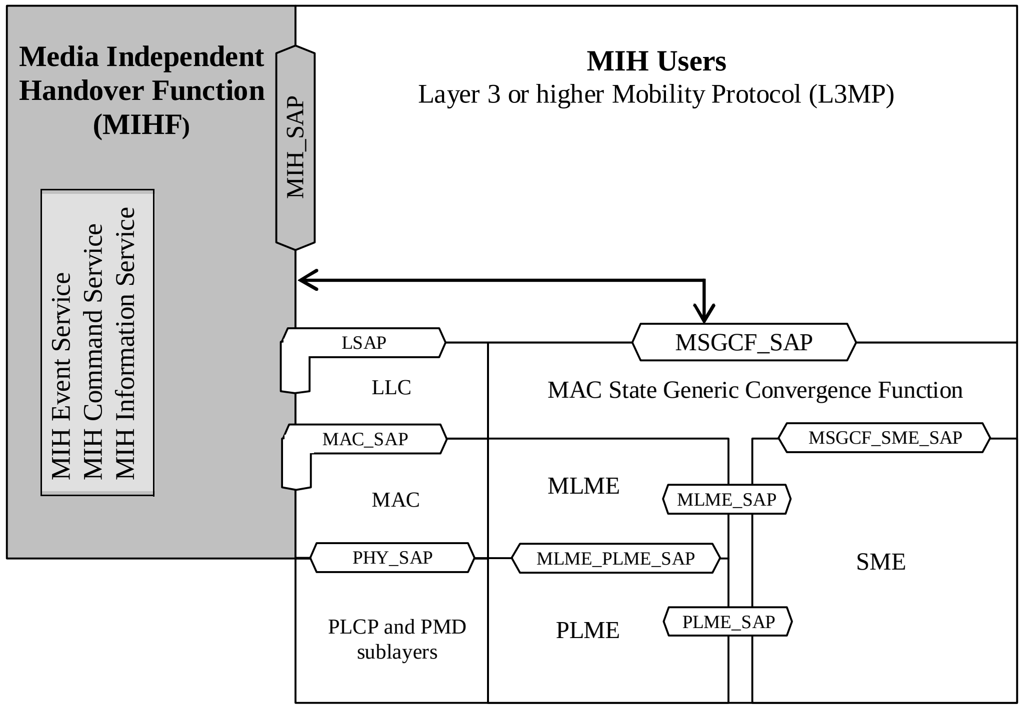


Figure 7 — MIH reference model for IEEE 802.11

5.5.5 MIHF reference model for IEEE 802.16

Figure 8 shows the MIHF for IEEE 802.16 based systems. The Management SAP (M_SAP) and Control SAP (C_SAP) are common between the MIHF and Network Control and Management System (NCMS).

The M_SAP specifies the interface between the MIHF and the management plane and allows MIHF payload to be encapsulated in management messages (such as MOB_MIH-MSG defined in [B21]). 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 MIHF and control plane. M_SAP and C_SAP also transport MIH messages to peer MIHF 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.

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

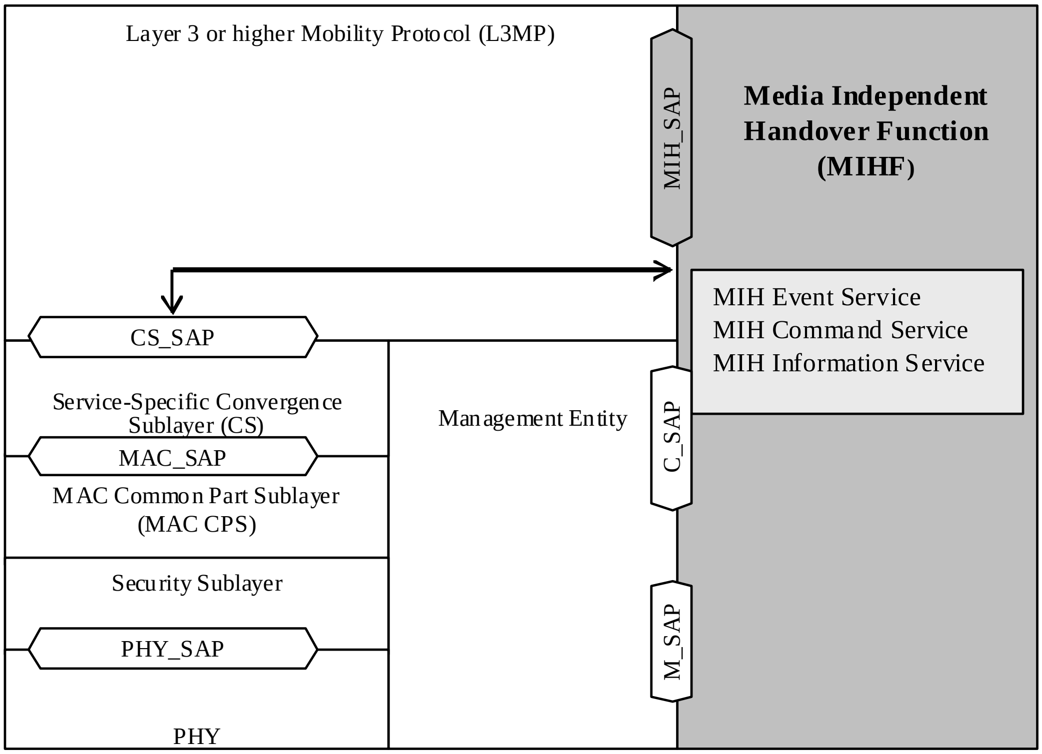


Figure 8 — MIH reference model for IEEE 802.16

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

5.5.6 MIHF reference model for 3GPP

Figure 9 illustrates the interaction between the MIHF and the 3GPP based systems. The MIHF services are specified by the MIH_3GLINK_SAP. However no new primitives or protocols need to be defined in the 3GPP specification for accessing these services. The MIHF services are mapped to existing 3GPP signaling functions (see Table J-3). The architectural placement of the MIHF is also decided by the 3GPP standard. Figure 9 is for illustrative purposes only and should not constrain implementations.

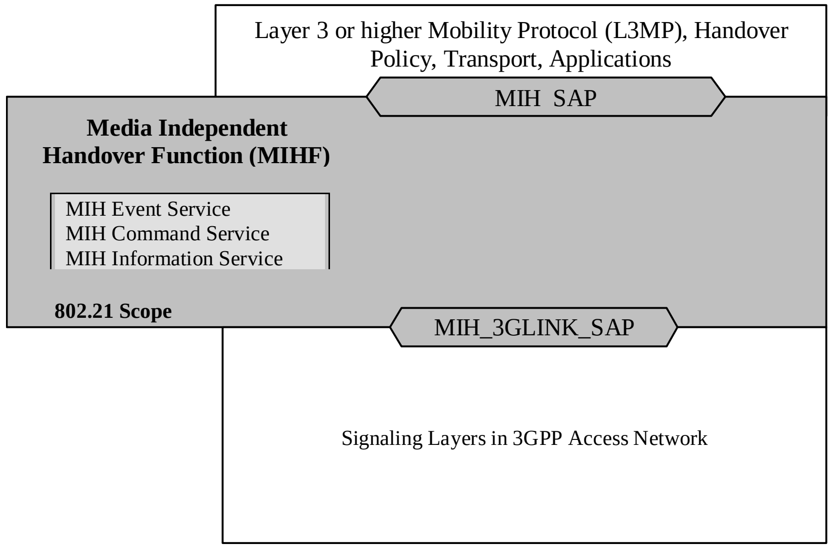


Figure 9 — MIH reference model for 3GPP systems

5.5.7 MIHF 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 MIH_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 Internet Engineering Task Force (IETF) request for comment (RFC) 1661 and 3GPP2 C.S0004-D is already established. Primitive information available from Upper Layer Signaling and Point-to-Point Protocol (PPP) can be directly used by mapping LAC SAP and PPP SAP primitives to IEEE 802.21 service primitives in order to generate an event.

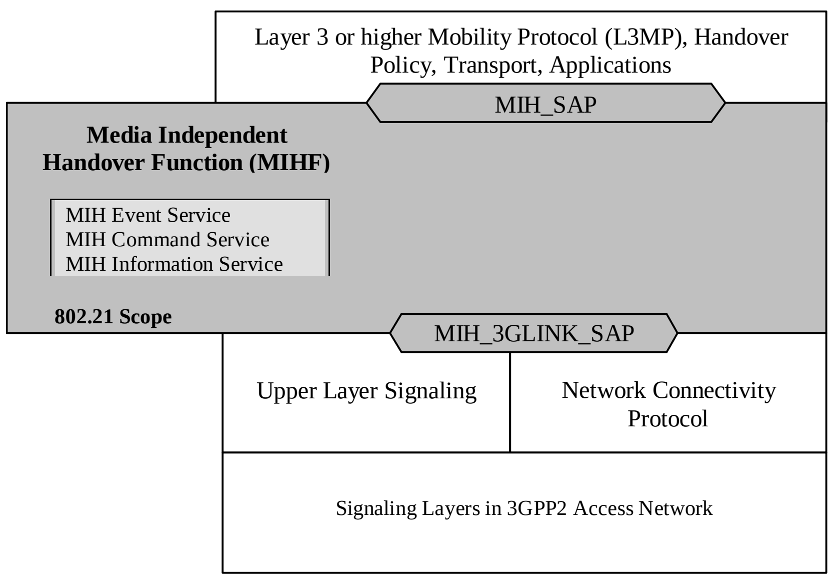


Figure 10 — MIH reference model for 3GPP2 systems

This mapping is illustrated in Table J-3, which provides an example of how 3GPP and 3GPP2 primitives can be 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” can be mapped and generated through the MIH_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 MIH_3GLINK_SAP as a Link_Up event.

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

5.6 Service access points (SAPs)

5.6.1 General

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

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

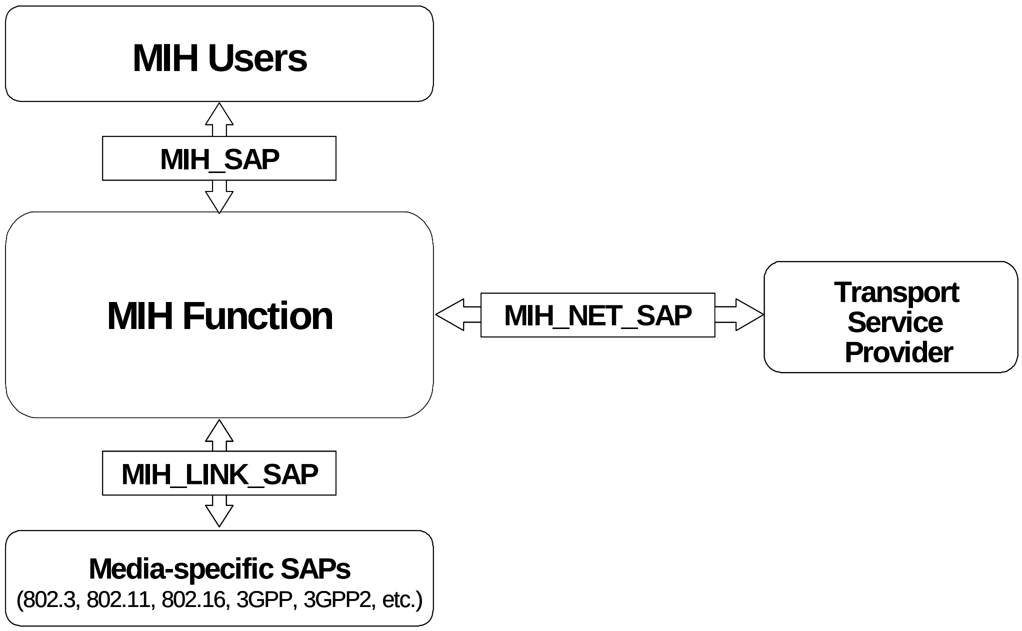


Figure 11—Relationship between different MIHF SAPs

Figure 11 shows the key MIHF-related SAPs for different networks:

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

b) The MIH_LINK_SAP specifies an abstract media dependent interface between the MIHF 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 MIH_LINK_SAP. Amendments are suggested to the respective media-specific SAPs to provide all the functionality as described by MIH_LINK_SAP;

c) The MIH_NET_SAP specifies an abstract media dependent interface of the MIHF that provides transport services over the data plane on the local node, supporting the exchange of MIH information and messages with remote MIHFs.

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 MIH_LINK_SAP maps to the technology-specific SAPs.

5.6.2.2 MIH_LINK_SAP

This SAP defines the abstract media dependent interface between MIHF 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 MIH_NET_SAP

MIH_NET_SAP defines the abstract media dependent interface of the MIHF that provides transport services over the data plane on the local node, supporting the exchange of MIH information and messages with remote MIHFs. For L2 this SAP uses the primitives provided by MIH_LINK_SAP.

5.6.2.4 MLME_SAP

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

5.6.2.5 C_SAP

The C_SAP, defined in the IEEE Std 802.16, provides the interface between the MIHF and the IEEE 802.16 control plane. This SAP is used for MIH exchanges between the MIHF and the lower layers of the management plane (as part of the IEEE 802.16 instantiation of the MIH_LINK_SAP).

5.6.2.6 M_SAP

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

5.6.2.7 MSGCF_SAP

This SAP, defined in the IEEE P802.11u/D2.0, provides services to MIHF based on the IEEE 802.11 MAC state machines and interactions between the IEEE 802.11 sublayers.

5.6.2.8 MIH_3GLINK_SAP

This SAP works as an umbrella that defines the interface between the MIHF 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 MIHF services, and hence no new primitives need to be defined in these specifications. Table J-3 lists this mapping.

5.6.2.9 LSAP

The LLC SAP (LSAP), defined in the IEEE Std 802.2, provides the interface between the MIHF and the Logical Link Control sublayer in IEEE 802.3 and IEEE 802.11 networks. This SAP is used for local MIH exchanges between the MIHF and the lower layers and for the L2 transport of MIH messages across IEEE 802 access links.

5.6.2.10 CS_SAP

The CS_SAP, defined in the IEEE Std 802.16, provides the interface between the MIHF and the service-specific Convergence Sublayer in IEEE 802.16 networks. This SAP is used for the L2 transport of MIH messages across IEEE 802.16 access links.

5.6.3 Media independent SAPs

5.6.3.1 MIH_SAP

This SAP defines the media independent interface between the MIHF and MIH Users such as an upper layer mobility protocol or a handover function that might reside at higher layers or a higher layer transport entity as well. The definition of this SAP is required to define the scope and functionality of the MIHF.

5.7 MIH protocol

5.7.1 General

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

5.7.2 Ethertype use and encoding

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

Table 2—MIH protocol ethernet type

Assignment	Value
MIH protocol ethernet type	To be assigned*

*NOTE --- Awaiting Ethertype assignment from IEEE Ethertype Registration Authority

5.7.3 Transport considerations

MIH 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 MIH 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 MIH 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 802.1X

The generic MAC service in both IEEE 802.3 network and 802.11 robust security network association (RSNA) networks (which use 802.1X REV 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 closed until authentication and association are successful.

The MIH messages that pass through the LSAP are distinguished from other protocols with an EtherType value of MIH protocol ethernet type in the LLC header.

For backward compatibility with an IEEE 802.11 wired equivalent privacy (WEP) network, which does not use 802.1X, MIH messages with EtherType value of MIH protocol ethernet type in the LLC header may go through the LSAP to use the MAC transport service in the data plane after successful completion of the WEP shared key authentication.

5.7.4.1 Port open state: LSAP transport

After successful authentication and association, the controlled port is open to the transport of authenticated messages. The MIH messages are then encapsulated into LLC protocol with EtherType value of MIH 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 MIHF protocol or to other protocols.

5.7.4.2 Port closed state

Until authentication has been completed, the controlled port is in closed state so that MIH messages cannot go through. However, in IEEE 802.11 and IEEE 802.16 networks, MIH message can be transported via the management plane prior to authentication .

5.7.5 802.16 using privacy and key management (PKMv2) and negotiated cipher suite

In an IEEE 802.16 network using privacy and key management (PKMv2) and negotiated cipher suite, message exchange between peer MIHF may be supported by MAC service through CS_SAP in addition to IP transport.

Before authentication and association, only MIH information service and MIH capability discovery messages can be transported using M_SAP via the management plane.

6. MIHF services

6.1 General

The MIHF provides the Media Independent Event Service, the Media Independent Command Service, and the Media Independent Information Service that facilitate handovers across heterogeneous networks. Clause 6 provides a general description of these services. These services are managed and configured through service management primitives, as discussed in 6.2.

6.2 Service management

6.2.1 General

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

- MIH Capability Discovery
- MIH Registration
- MIH Event Subscription

In order to know the services that are supported by an MIH peer, the MIH node performs MIH Capability Discovery. The MIH node performs MIH Capability Discovery with different MIH 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 can be invoked by a local MIH User, a remote MIH User, or both, respectively.

Table 3—Service management primitives

Service management primitive	(L) ocal, (R) emote	Defined in	Comments
MIH_Capability_Discover	L, R	7.4.1	Discover the capabilities of a local or remote MIHF.
MIH_Register	R	7.4.2	Register with a remote MIHF.
MIH_DeRegister	R	7.4.3	Deregister from a remote MIHF.
MIH_Event_Subscribe	L, R	7.4.4	Subscribe for one or more MIH events with a local or remote MIHF.
MIH_Event_Unsubscribe	L, R	7.4.5	Unsubscribe for one or more MIH events from a local or remote MIHF.

6.2.3 MIH capability discovery

The MIH Capability Discovery procedure is used by an MIH User to discover a local or remote MIHF's capabilities in terms of MIH Services (Event Service, Command Service, and Information Service). MIH

Capability Discovery is performed either through the MIH protocol or through media-specific Layer 2 broadcast messages (i.e., IEEE 802.11 Beacon frames, IEEE 802.16 DCD).

6.2.4 MIH registration

MIH registration provides a mechanism for an MIH entity to make its presence known to a peer MIH entity. For example, in a network controlled inter-technology handover framework, MIH registration can be used by an MN to declare its presence to a selected MIH PoS.

The use of MIH Registration is optional. In other words, an MIH entity is allowed to provide MIH Services to a peer MIH entity without requiring an MIH Registration from that peer.

6.2.5 MIH event subscription

The MIH Event Subscription mechanism allows an MIH User to subscribe for a particular set of events that originates from a local or remote MIHF. See 6.3.2 for a more detailed description of MIH 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 MIH information and messages between the local and remote MIHF. For transport services over L2, MIH_NET_SAP utilizes the primitives specified by the MIH_LINK_SAP. For transport services over L3 the primitives are specified by MIH_NET_SAP. Please refer to 7.5 for more details on MIH_NET_SAP.

6.3 Media independent event service

6.3.1 Introduction

In general, handovers can be initiated either by the MN or by the network. Events relevant to handover originate from MAC, PHY or MIHF 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 MIHF 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 MIH Users and MIHFs 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. Higher layer protocols and other entities can take a more cautious approach when events originate remotely as opposed to when they originate locally. These events can also be used for horizontal handovers.

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

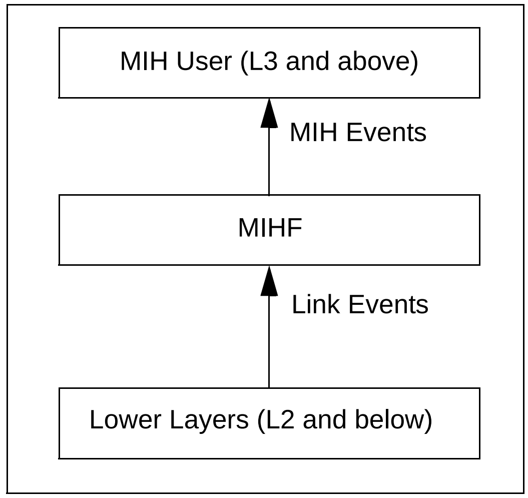


Figure 12 — Link events and MIH events

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

All Link Events are local in nature and propagate from the local lower layer to the local MIHF. MIH Events are local or remote. A remote MIH Event traverses the medium from a remote MIHF to the local MIHF and is then dispatched to local MIH Users that have subscribed to this remote event, as shown in Figure 13.

A Link Event that is received by the MIHF can also be sent to a remote MIH entity as a remote MIH Event.

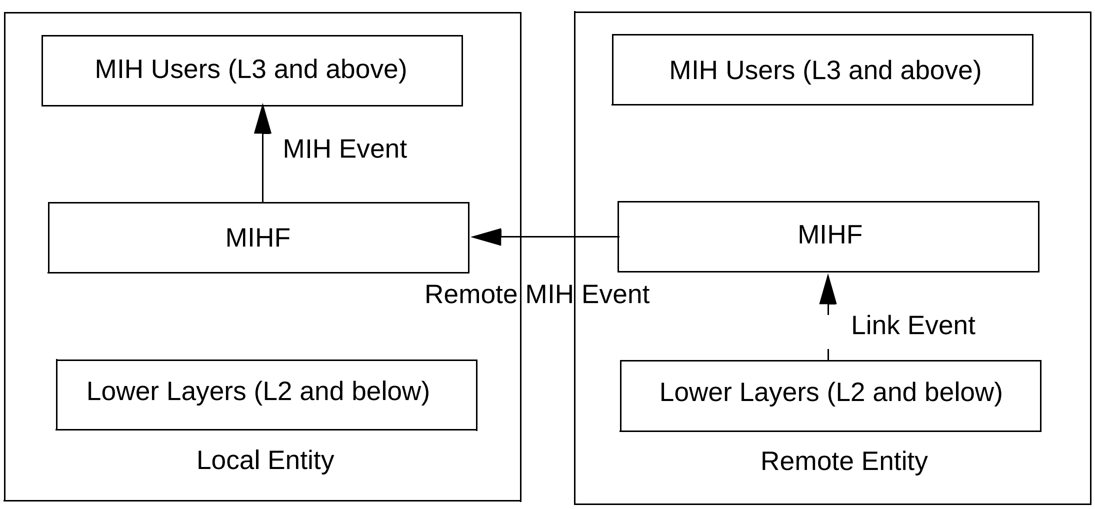


Figure 13 — Remote MIH events

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 can be divided into Link Events Subscription and MIH Events Subscription. Link Events Subscription is performed by the MIHF with the event source entities in order to determine the events that each event source (link) is able to provide. MIH Event Subscription is performed by upper layer entities with the MIHF 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 MIHF.

6.3.2.2 Link events subscription

During initialization the MIHF 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 MIHF subscribes individually with each of these link layers based on user preferences.

6.3.2.3 MIH events subscription

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

6.3.3 Event service flow model

Figure 14 shows the event flow model for link events and MIH events.

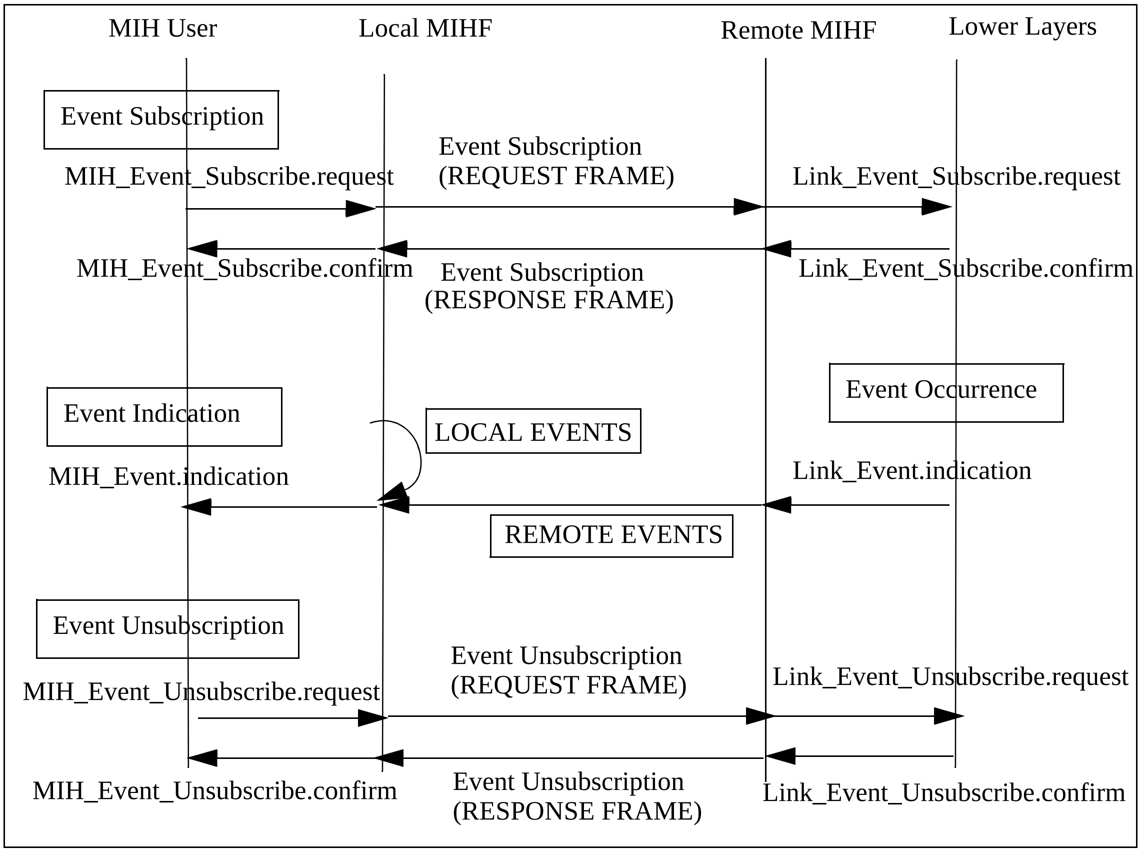


Figure 14 — MIH events subscription and flow

6.3.4 Link events

6.3.4.1 General

The Media Independent Event Service supports 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) network indicates a loss of link connectivity in the near future.

d) **Link Handover events**: These events inform upper layers about the occurrence of L2 handovers/ link switches if supported by the given media type.[1]

e) **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 manage-

[1]The mechanism that triggers and executes a link layer handover/switch (also referred as an L2 handover) is specified within the corresponding media-specific standard and out of scope of this standard.

ment 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 will result 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., unacked PDUs). These buffered PDUs will be 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 will 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, i.e., it is not associated with other events with a state machine.	7.3.2
Link_Down	State Change	L2 connection is broken and link is not available for use. This event is a discrete event, i.e., it is not associated with other events with a state machine.	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_Handover_Imminent	Link Handover	L2 handover is imminent based on changes in link conditions.	7.3.6
Link_Handover_Complete	Link Handover	L2 link handover to a new PoA has been completed.	7.3.7
Link_PDU_Transmit_Status	Link Transmission	Indicate transmission status of a PDU.	7.3.8

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 on resort to inter-technology handovers.

6.3.5 MIH events

Table 5 defines MIH Events. An MIH event is marked as local only (L), remote only (R), or local and remote (L, R), indicating whether it can be subscribed by a local MIH User, a remote MIH User, or both, respectively.

Table 5—MIH events

MIH event name	(L) ocal (R) emote	Description	Defined in
MIH_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
MIH_Link_Up	L, R	L2 connection is established and link is available for use.	7.4.7
MIH_Link_Down	L, R	L2 connection is broken and link is not available for use.	7.4.8
MIH_Link_Parameters_Report	L, R	Link parameters have crossed a specified threshold and need to be reported.	7.4.9
MIH_Link_Going_Down	L, R	Link conditions are degrading and connection loss is imminent.	7.4.10
MIH_Link_Handover_Imminent	L, R	L2 handover is imminent based on either the changes in the link conditions or additional information available in the network. For example, the network decides that an application requires a specific QoS that can be best provided by a certain access technology.	7.4.11
MIH_Link_Handover_Complete	L, R	L2 link handover to a new PoA has been completed.	7.4.12
MIH_Link_PDU_Transmit_Status	L	Indicate transmission status of a PDU.	7.4.13

6.3.6 Interaction between MIH 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 a MN.

During heterogeneous handovers a MN can switch from one link technology to another. This will result in a change in the PoA that the MN is connected to. The target PoA and the source PoA may or may not 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 can update its L2 address to IP address mapping.

Link layer triggers such as Link Going Down and Link Up can be 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 can also benefit from triggers such as

Link Going Down. Timely receipt of such triggers by the AR in case of network-controlled handovers can enable MIP signaling to establish the new route to take place in parallel with other handover message exchange and can thus reduce 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 MIH Users to the lower layers in the reference model. MIH Users utilizes command services to determine the status of links and/or control the multi-mode device for optimal performance. Command services also enable MIH Users to facilitate optimal handover policies. For example, the network initiates and control handovers to balance the load of two different access networks.

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 MIH 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: MIH Commands and Link Commands. Figure 15 shows link commands and MIH commands.

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

Link Commands originate from the MIHF 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 MIHF, these commands are executed on behalf of the MIH Users.

The MIH commands are generated by the MIH Users and sent to the MIHF. MIH commands can be local or remote. Local MIH commands are sent by MIH Users to the MIHF in the local protocol stack.

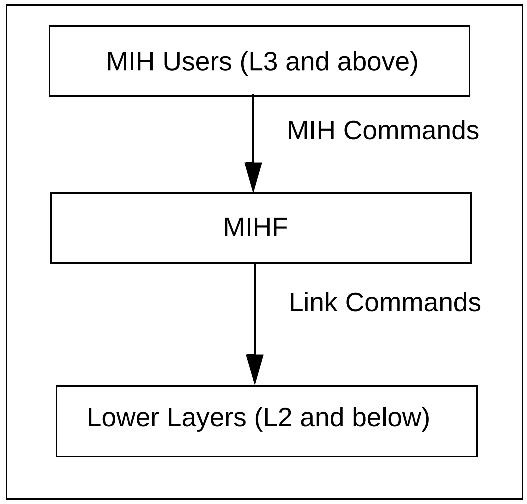


Figure 15 — Link commands and MIH commands

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

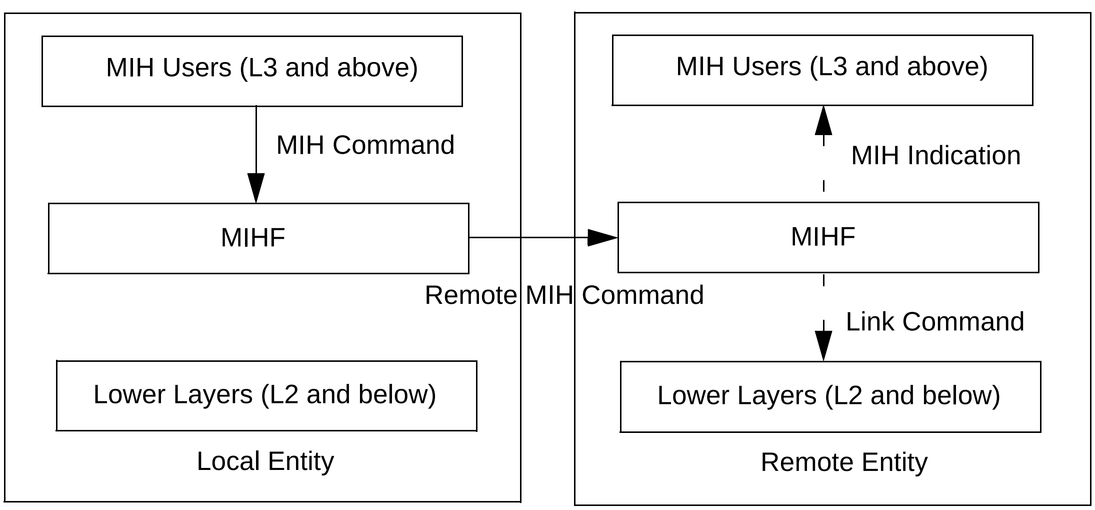
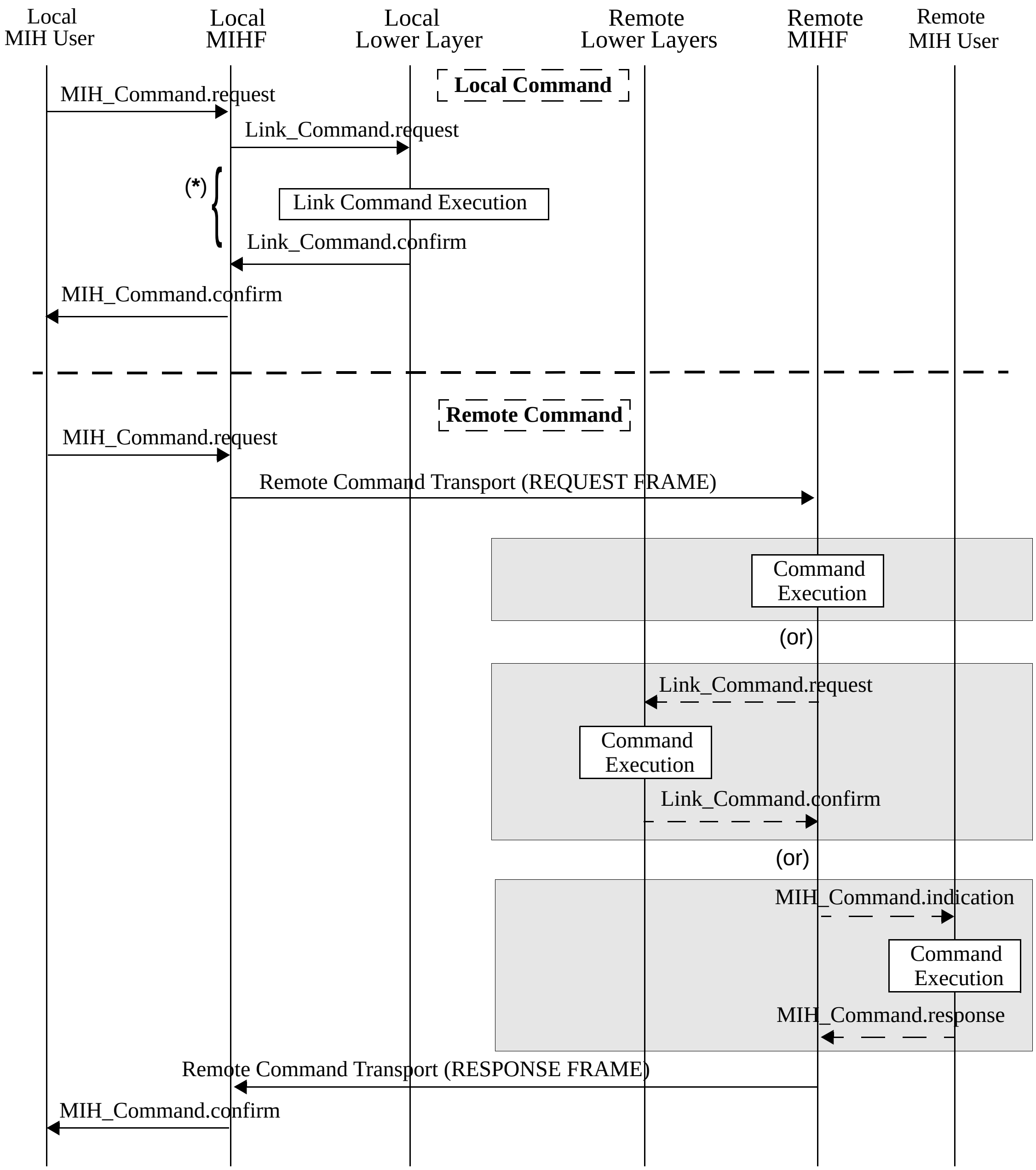


Figure 16 — Remote MIH command

6.4.2 Command service flow model



(*) There might be no corresponding Link_Command primitives, and one or more media-specific link primitives can be used here.

Figure 17 — Command service flow

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 L. Remote commands are transported over network layer protocols or link layer protocols.

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.	7.3.9
Link_Event_Subscribe	Subscribe to one or more events from a link.	7.3.10
Link_Event_Unsubscribe	Unsubscribe from a set of link layer events.	7.3.11
Link_Get_Parameters	Get parameters measured by the active link, such as signal-to-noise ratio (SNR), BER, received signal strength indication (RSSI).	7.3.12
Link_Configure_Thresholds	Configure thresholds for Link Parameters Report event.	7.3.13
Link_Action	Request an action on a link layer connection	7.3.14

6.4.3.2 MIH commands

6.4.3.2.1 General

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

Table 7—MIH commands

MIH command	(L) ocal, (R) emote	Comments	Defined in
MIH_Link_Get_Parameters	L, R	Get the status of a link.	7.4.14
MIH_Link_Configure_Thresholds	L, R	Configure link parameter thresholds	7.4.15
MIH_Link_Actions	L, R	Control the behavior of a set of links	7.4.16
MIH_Net_HO_Candidate_Query	R	Network initiates handover and send a list of suggested networks and associated Points of Attachment.	7.4.17
MIH_MN_HO_Candidate_Query	R	Command used by MN to query and obtain handover related information about possible candidate networks.	7.4.18

Table 7—MIH commands

MIH command	(L) ocal, (R) emote	Comments	Defined in
MIH_N2N_HO_Query_Resources	R	This command is sent by the serving MIHF entity to the target MIHF entity to allow for resource query.	7.4.19
MIH_MN_HO_Commit	R	Command used by MN to notify the serving network of the decided target network information.	7.4.20
MIH_Net_HO_Commit	R	In this case the network commits to do the handover and sends the choice of selected network and associated PoA.	7.4.21
MIH_N2N_HO_Commit	R	Command used by a serving network to inform a target network that an MN is about to move toward that network, initiate context transfer (if applicable), and perform handover preparation.	7.4.22
MIH_MN_HO_Complete	R	Notification from MIHF of the MN to the target or source MIHF indicating the status of handover completion.	7.4.23
MIH_N2N_HO_Complete	R	Notification from either source or target MIHF to the other (i.e. peer) MIHF indicating the status of the handover completion.	7.4.24

6.4.3.2.2 Naming convention for MIH handover commands

Generally, there are three types of MIH handover command primitives based on the functionality specified for the following scenarios: i) MN to Network, ii) Network to MN, and iii) Network to Network. This classification helps to ensure the specification of the proper protocol functionality and the relevant parameters for specific use as determined by the origination and the destination points.

Accordingly, these commands have a naming convention that identifies the origination point in the primitive name, as shown in Table 8. This convention is followed by the MIHF to ensure that these commands are utilized for the intended purpose. The destination point applies for remote commands only.

Table 8—Naming convention for MIH handover command primitives

Primitive name prefix	Originating point	Destination point
MIH_MN_HO_***	MN	Network
MIH_Net_HO_***	Network	MN
MIH_N2N_HO_***	Network	Network

6.4.3.2.3 Mobile initiated handovers

In this case the MN initiates the handovers. The network selection policy function in this case resides on the mobile node. The MN directly uses the set of MIH_MN_HO_*** commands and may indirectly cause some MIH_N2N_HO_*** commands to be used when initiating handovers. The MN can use these commands to query the list of available candidate networks, reserve any required resources at the candidate target network and indicate the status of handover operation to the MIHF in the network.

6.4.3.2.4 Network initiated handovers

In this case the network initiates the handovers. The network selection policy function in this case resides on the network. The network uses the set of MIH_Net_HO_*** in conjunction with any MIH_N2N_HO_*** commands for initiating handovers. The network can use these commands to query the list of resources currently being used by the MN, the serving network can reserve any required resources at the candidate target network and the network can command the MN to commit to performing a handover to a specific network.

6.5 Media independent information service

6.5.1 Introduction

Media Independent Information Service (MIIS) provides a framework by which an MIHF, residing in the MN or in the network, discovers and obtain 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 roaming across these networks.

Media Independent Information Service includes support for various Information Elements (IEs). Information Elements 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 can make 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), 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 helps 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 can be provided as part of the Information Service. For example in certain cases classification of different networks into categories such as public, enterprise, home, and others influences a handover decision. Other information here 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 (MN) and a certain MIHF in the network. The MIHF in the MN discovers the MIH capability support from the PoA through the media-specific broadcast information containing the system capabilities. It should be noted that the pre-authentication query facility is provided only for MIH information query and cannot be used for carrying other MIH protocol services except MIES and/or MICS capability discovery query using MIH_Capability_Discover embedded into L2 management frames. Additionally, any MIHF within the network can request for the set of information elements from a peer MIHF located in the same or a different network using the MIH 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 MIH 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 will be in the order of significance to the client. In case the query results exceed the maximum response size, the least significant query results will 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 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 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) Other information that is access network specific, service specific, or vendor/network specific.

Table 9 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 9—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 10 represents the list of Information Elements and their semantics. Each Information Element has an abstract data type (see C.3.8 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 10. Vendors and network operators can implement new IEs using the Vendor Specific IEs. These IEs will then be available only in vendor or operator specific deployments.

Table 10—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

Table 10—Information elements

Name of information element	Description	Data type
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_MOB_MGMT_PROT	Type of mobility management protocol supported	IP_MOB_MGMT
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
IE_NET_MOBILE_NETWORK	Indicator whether the access network is a mobile network.	BOOLEAN
PoA Specific Information Elements		
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
PoA Specific Higher Layer Service Information Elements		
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

Table 10—Information elements

Name of information element	Description	Data type
Other Information Elements		
Vendor specific IEs	Vendor Specific Services	N/A

In certain access network deployments, some PoA properties (e.g., data rate, IP configuration methods, 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 can be providing support for a particular network. Thus support for IEEE 802.11 network is provided by both Operator_1 and Operator_2. A single operator can provide support for multiple networks. Thus 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.

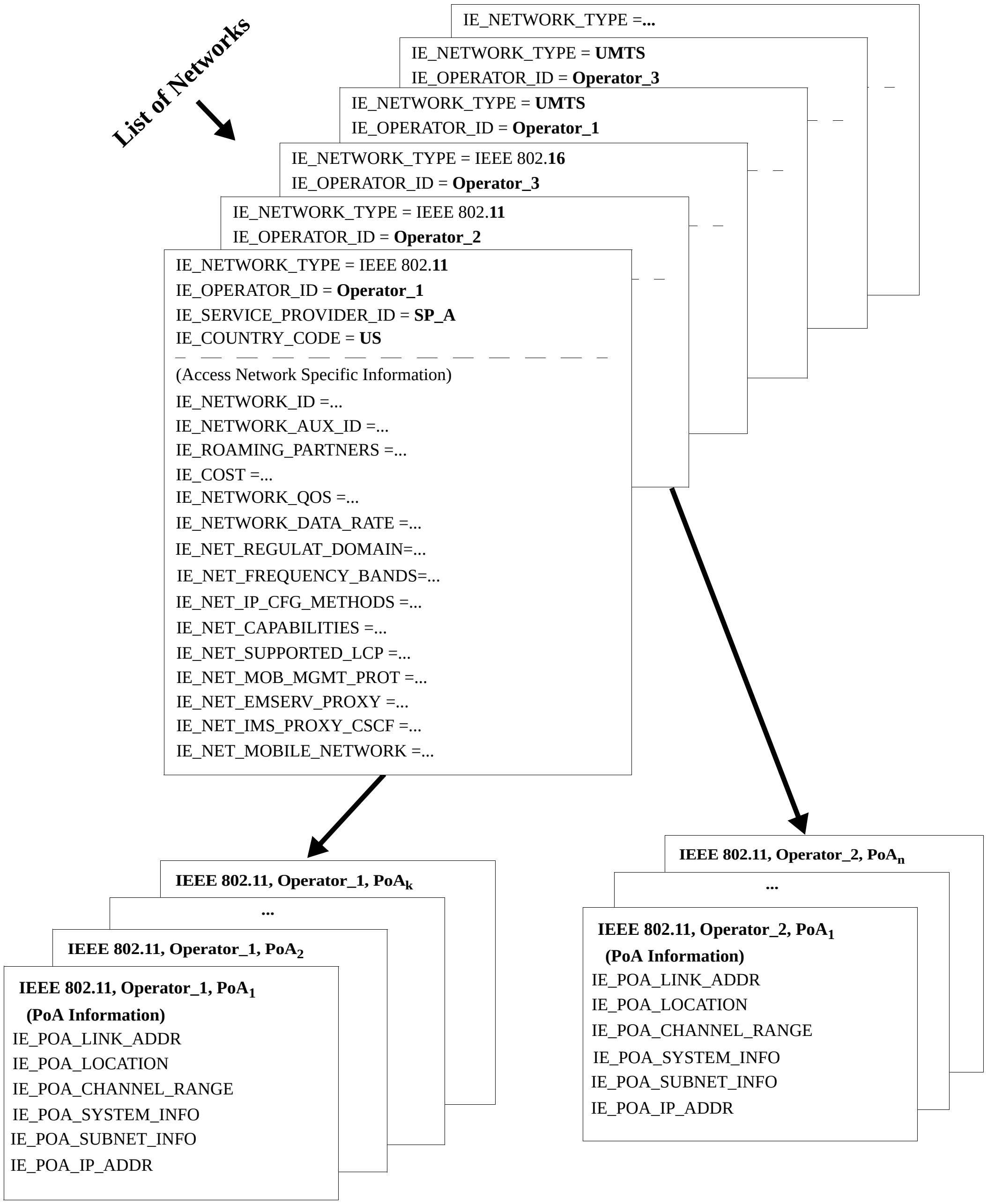


Figure 18—Depicting a list of neighboring networks with information elements.

6.5.5 Definition of information element namespace

Each Information Element ID is a 32 bit value. Table 11 defines the Information Element namespace. The IEEE 802.21 specific Information Elements are assigned identifiers as per this standard. Please refer to Table B-1 (in Annex B) for more details. Vendors specify their own IEs using the name space allocated to them. A set of IE name space 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 11—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 B-1 in Annex B.
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). 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

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 form as shown in Figure 19.

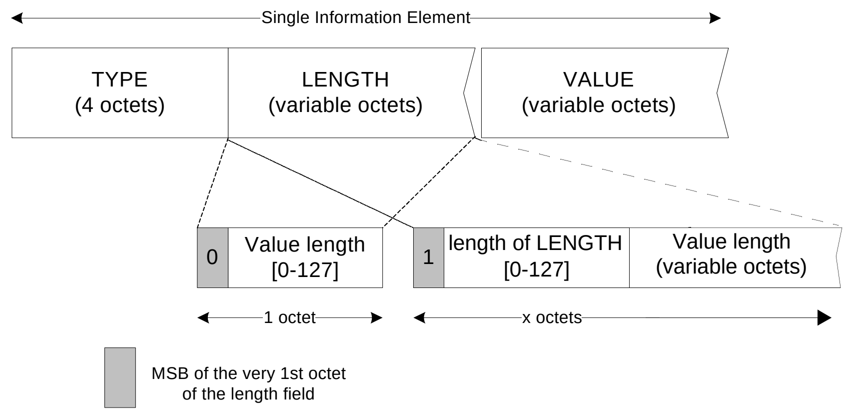


Figure 19 — TLV representation of information elements

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 2nd 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 12.

  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 12—IE_CONTAINER_LIST_OF_NETWORKS definition

Type= (see Table B-1)	**Length=** *variable*
IE_CONTAINER_NETWORK #1	

Table 12—IE_CONTAINER_LIST_OF_NETWORKS definition

IE_CONTAINER_NETWORK #2 (optional)
…
IE_CONTAINER_NETWORK #k (optional)
Vendor Specific IE (optional)

- **IE_CONTAINER_NETWORK** - contains all the information depicting an access network, as shown in Table 13.

  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 decisions

Table 13—IE_CONTAINER_NETWORK definition

Type= (see Table B-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_MOB_MGMT_PROT (optional)	
IE_NET_EMSERV_PROXY (optional)	
IE_NET_IMS_PROXY_CSCF (optional)	
IE_NET_MOBILE_NETWORK (optional)	
IE_CONTAINER_POA #1 (optional)	
IE_CONTAINER_POA #2 (optional)	

Table 13—IE_CONTAINER_NETWORK definition

...
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 14:

Table 14—IE_CONTAINER_POA definition

Type= (see Table B-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 C.

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 C-15) can be useful 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 will 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 Table C-3) and LOCATION value (defined in Table C-10) in the same query. Moreover, the NGHB_RADIUS value (defined in Table C-15), if provided, indicates the radius of the neighborhood, centered at the indicated location, within which all available access networks will be included in the list of neighboring networks. If NGHB_RADIUS value is not present, the Information Server will 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 C-15 for definition) can be 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 C-15 for definition) can be 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 C-15 for definition) can be 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.

The following rules are followed for generating returned IEs:

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

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 MIH_Get_Information response message.
c) If a 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 MIH_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 will 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 MIHF that implements the MIIS to support flexible and efficient information queries.

6.5.7.2 The 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 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 will always use the updated version of extended schema as opposed to using the pre-provisioned version.

The basic schema is defined in Annex E. The basic schema contains the schema for information elements defined in Table 10. 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.

6.5.8 Information service flow

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

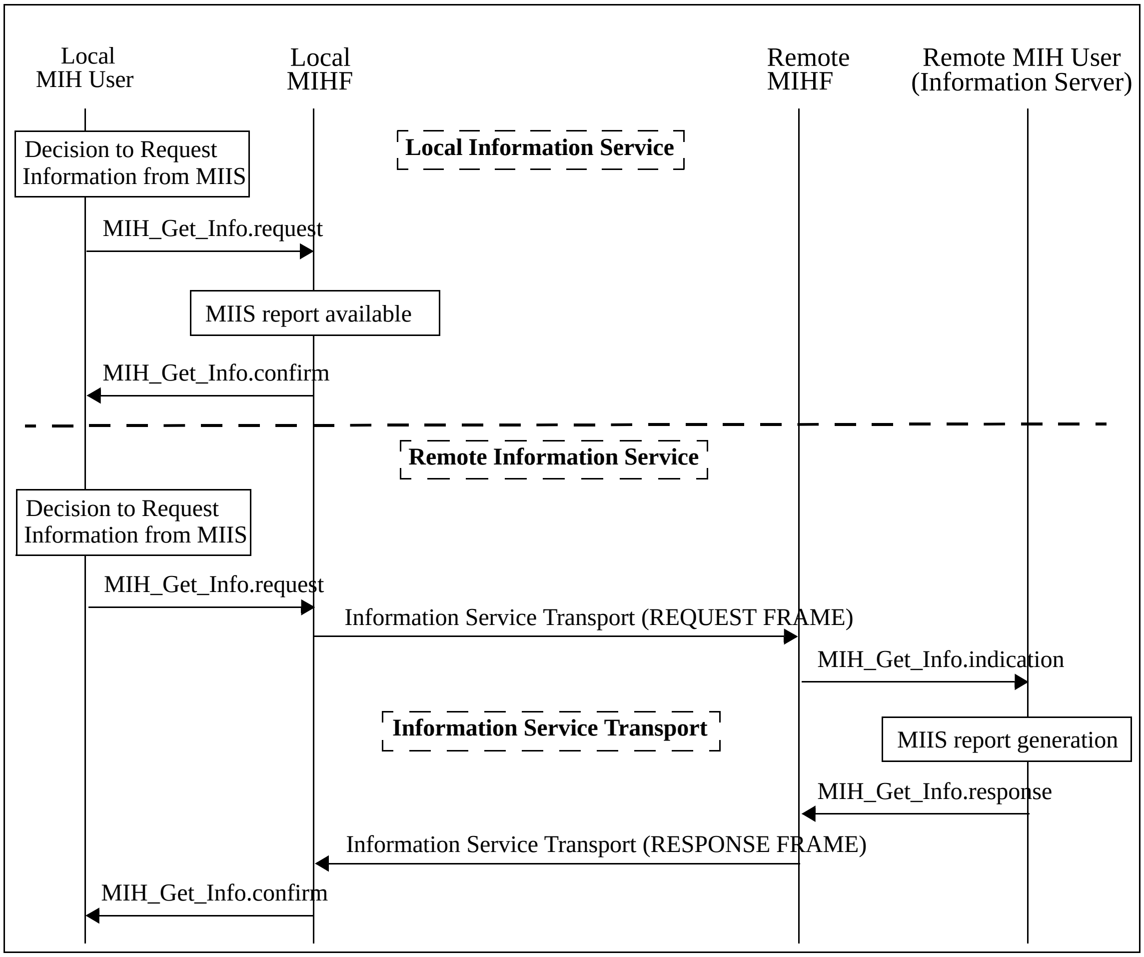


Figure 20 — MIIS information flow

7. Service access points (SAPs) and primitives

7.1 Introduction

The MIH Function uses the following SAPs for interfacing with other entities.

Media dependent SAPs:

a) MIH_LINK_SAP: Abstract media dependent interface of MIHF with the lower layers of the media-specific protocol stacks. The mappings between MIH_LINK_SAP and various media-specific SAPs are described in Annex J.2.

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

Media independent SAPs:

— MIH_SAP: This SAP defines the media independent interface between the MIHF and MIH 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 C.

7.2.2 Media dependent SAPs

7.2.2.1 MIH_LINK_SAP

The primitives defined as part of the MIH_LINK_SAP are described in Table 15. Annex J contains their mapping to several specific link technologies.

Table 15—MIH_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_Handover_Imminent	Event	L2 handover is imminent	7.3.6
Link_Handover_Complete	Event	L2 handover has been completed	7.3.7

Table 15—MIH_LINK_SAP primitives

Primitives	Service category	Description	Defined in
Link_PDU_Transmit_Status	Event	Indicate transmission status of a PDU	7.3.8
Link_Capability_Discover	Command	Query and discover the list of supported link layer events and link layer commands	7.3.9
Link_Event_Subscribe	Command	Subscribe for event notifications	7.3.10
Link_Event_Unsubscribe	Command	Unsubscribe from event notifications	7.3.11
Link_Get_Parameters	Command	Request parameters of medium	7.3.12
Link_Configure_Thresholds	Command	Configure link thresholds for Link events	7.3.13
Link_Action	Command	Request an action on a link layer connection	7.3.14

7.2.2.2 MIH_NET_SAP

The primitive defined for MIH_NET_SAP is described in Table 16.

Table 16—MIH_NET_SAP primitive

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

7.2.3 Media independent SAPs

7.2.3.1 MIH_SAP

The primitives defined as part of MIH_SAP are described in Table 17.

Table 17—MIH_SAP primitives

Primitives	Service category	Description	Defined in
MIH_Capability_Discover	Service management	Discover list of Events and Commands supported by MIHF.	7.4.1
MIH_Register	Service management	Register with a remote MIHF.	7.4.2
MIH_DeRegister	Service management	Deregister with a remote MIHF.	7.4.3
MIH_Event_Subscribe	Service management	Subscribe for MIH event notifications	7.4.4
MIH_Event_Unsubscribe	Service management	Unsubscribe from MIH event notifications	7.4.5
MIH_Link_Detected	Event	A new link is detected	7.4.6

Table 17—MIH_SAP primitives

Primitives	Service category	Description	Defined in
MIH_Link_Up	Event	L2 connection has been established	7.4.7
MIH_Link_Down	Event	L2 connectivity is lost	7.4.8
MIH_Link_Parameters_Report	Event	Link parameters have crossed specified threshold	7.4.9
MIH_Link_Going_Down	Event	L2 connectivity is predicted to go down	7.4.10
MIH_Link_Handover_Imminent	Event	L2 handover is imminent	7.4.11
MIH_Link_Handover_Complete	Event	L2 handover has been completed	7.4.12
MIH_Link_PDU_Transmit_Status	Event	Indicate transmission status of a PDU	7.4.13
MIH_Link_Get_Parameters	Command	Get the status of link	7.4.14
MIH_Link_Configure_Thresholds	Command	Configure link parameter thresholds	7.4.15
MIH_Link_Actions	Command	Control the behavior of a set of links	7.4.16
MIH_Net_HO_Candidate_Query	Command	Initiate handover	7.4.17
MIH_MN_HO_Candidate_Query	Command	Initiate MN query request for candidate network	7.4.18
MIH_N2N_HO_Query_Resources	Command	Query available network resources	7.4.19
MIH_MN_HO_Commit	Command	Notify the serving network of the decided target network information	7.4.20
MIH_Net_HO_Commit	Command	Network has committed to handover	7.4.21
MIH_N2N_HO_Commit	Command	Notify target network that the serving network has committed to handover.	7.4.22
MIH_MN_HO_Complete	Command	Initiate MN handover complete notification	7.4.23
MIH_N2N_HO_Complete	Command	Handover has been completed	7.4.24
MIH_Get_Information	Information	Request to get information from repository	7.4.25

MIH command primitives defined in MIH_SAP indicates their destination as either the local MIHF or a remote MIHF. For the remote case, the local MIHF will first process the primitive to create an MIH message and then forward the message to the destination peer MIHF 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 C.2.

7.3 MIH_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 will be able to establish connectivity with the detected link, but just that the MN can attempt to gain connectivity. MIH Users and the MIHF evaluate additional properties of the link before attempting to establish a L2 connection with the link. Moreover, Link_Detected is not generated when additional PoAs of the same link are discovered. In case of 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	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 MIHF receives this event from the link layer. The MIHF shall pass this notification to the MIH User(s) that have subscribed for this notification. The MIH User(s), including the MIHF 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 of time.

7.3.2.2 Semantics of service primitive

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

Parameters:

Name	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.
MobilityManagementSupport	IP_MOB_MGMT	(Optional) Indicates the type of Mobility Management Protocol supported by the new PoA.

7.3.2.3 When generated

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

7.3.2.4 Effect on receipt

The MIHF shall pass this link notification to the MIH User(s) that has subscribed for this notification in an MIH_Link_Up event. The MIH 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	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 is generated when layer 2 connectivity is lost. Layer 2 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 retransmitted packets along with loss of reception of broadcast frames.

7.3.3.4 Effect on receipt

The MIHF passes this link notification to the MIH User(s) that has subscribed for this notification in an MIH_Link_Down event. The MIH 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.indication

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

7.3.4.2 Semantics of service primitive

```
Link_Parameters_Report.indication(
                                   LinkIdentifier,
                                   LinkParametersReportList
                                   )
```

Parameters:

Name	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.

The primitive to set parameter thresholds that could trigger this event is specified in 7.3.13.

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 MIHF receives this event from the link layer. The MIHF then passes this notification to the MIH User(s) that have subscribed for this notification. The MIH 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 MIHF collectively evaluates different parameters and give 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 can be the indication to initiate handover procedures.

7.3.5.2 Semantics of service primitive

```
Link_Going_Down.indication      (
                                LinkIdentifier,
                                TimeInterval,
                                LinkGoingDownReason
                                )
```

Parameters:

Name	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 case of 802.11 networks, this notification is generated when the established 802.11 network connection is expected to go down within the specified time interval by the 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 can also be expected as the result of a predictive algorithm that monitors link quality. The details of such a predictive algorithm used are beyond the scope of this standard. This event is not generated when the 802.11 station (STA) transitions from one AP to another in the same network.

7.3.5.4 Effect on receipt

The MIHF receives this event from the link layer. The MIHF then passes this notification to the MIH User(s) that have subscribed for this notification. MIH User(s) takes different actions on this notification. MIH Users prepare to initiate handovers.

7.3.6 Link_Handover_Imminent.indication

7.3.6.1 Function

Link_Handover_Imminent is generated when a native link layer handover or switch decision has been made and its execution is imminent (as opposed to Link_Going_Down that only indicates that a link is losing connectivity due to a change in a certain link condition such as signal strength, but does not guarantee that a link switch-over has been decided by the link layer itself). It contains information about the new point of attachment of the MN (the LinkIdentifier parameter contains information about the new PoA). This is a Link Handover event as discussed in 6.3.4.1.

7.3.6.2 Semantics of service primitive

```
Link_Handover_Imminent.indication        (
                                          Old Link Identifier,
                                          New Link Identifier,
                                          OldAccessRouter,
                                          NewAccessRouter
                                          )
```

Parameters:

Name	Type	Description
OldLinkIdentifier	LINK_TUPLE_ID	Identifier of the old link.
NewLinkIdentifier	LINK_TUPLE_ID	Identifier of the new link.
OldAccessRouter	LINK_ADDR	(Optional) Link address of old Access Router.
NewAccessRouter	LINK_ADDR	(Optional) Link address of new Access Router.

7.3.6.3 When generated

Depending on whether it is the MN or the Network, it is generated when a native link layer handover or switch decision has been made and its execution is imminent.

7.3.6.4 Effect on receipt

The MIHF receives this event from the link layer. The MIHF then passes this notification to the MIH User(s) that have subscribed for this notification. The MIH User(s) takes necessary actions to minimize the effect of the pending native link layer handover or switch on user data transfer. This event is also used as an indication to start buffering packets.

7.3.7 Link_Handover_Complete.indication

7.3.7.1 Function

Link_Handover_Complete event is generated whenever a native link layer handover/switch has just been completed (as opposed to Link_Up that only indicates that a link has been brought up for L2 connectivity, but does not indicate that a native link handover/switch-over has just been completed by the link layer). Notifying the upper layer of this event improves transport, session and application layer responsiveness to the link changes. They can better adapt their data flows by resuming flows upon receiving this indication. The upper layers also use this event to check whether their IP configuration needs to be updated. This is a link layer event that exists for intra-technology handovers defined in many media types. This event is applicable for the MN only and is valid only for intra-technology handovers. This is a Link Handover event as discussed in 6.3.4.1.

7.3.7.2 Semantics of service primitive

```
Link_Handover_Complete.indication        (
                                          OldLinkIdentifier,
```

NewLinkIdentifier,
OldAccessRouter,
NewAccessRouter,
LinkHandoverStatus
)

Parameters:

Name	Type	Description
OldLinkIdentifier	LINK_TUPLE_ID	Identifier of the old link
NewLinkIdentifier	LINK_TUPLE_ID	Identifier of the new link
OldAccessRouter	LINK_ADDR	(Optional) Link address of old Access Router
NewAccessRouter	LINK_ADDR	(Optional) Link address of new Access Router
LinkHandoverStatus	STATUS	Status of the link handover.

7.3.7.3 When generated

This is generated whenever a L2 link layer handover or switch has just been completed.

7.3.7.4 Effect on receipt

The MIHF receives this event from the link layer. The MIHF then passes this notification to the MIH User(s) that have subscribed for this notification. Upon reception of this event, an upper layer stops any handover adaptation that it has engaged to cope with the just completed native link layer handover/switch and resume normal data transfer. This event is also used as an indication that a re-verification of the IP parameter should be considered.

7.3.8 Link_PDU_Transmit_Status.indication

7.3.8.1 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 will be decided by the upper layer (i.e., the MIH User that subscribes to this event). The MIHF and link layer just pass and return the Packet Identifier and do not need to understand its syntax and semantics.

To avoid receiving excessive amount of link PDU transmission status indications, an MIH 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 MIH_Link_Actions.request to perform a handover, and to unsubscribe from the event once it receives indication that the handover is completed.

7.3.8.2 Semantics of service primitive

Link_PDU_Transmit_Status.indication (
LinkIdentifier,
PacketIdentifier,
TransmissionStatus
)

Parameters:

Name	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

7.3.8.3 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.

7.3.8.4 Effect on receipt

The MIHF receives this event from the link layer. The MIHF then passes this notification to the MIH User(s) that have subscribed for this notification. The MIH User(s) takes different actions on this notification. A higher layer intermediate buffer management entity in MIH 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.

7.3.9 Link_Capability_Discover

7.3.9.1 Link_Capability_Discover.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 link layer commands.

7.3.9.1.2 Semantics of service primitive

No primitive parameters exist for this primitive.

Link_Capability_Discover.request ()

7.3.9.1.3 When generated

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

7.3.9.1.4 Effect on receipt

The recipient responds immediately with Link_Capability_Discover.confirm primitive.

7.3.9.2 Link_Capability_Discover.confirm

7.3.9.2.1 Function

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

7.3.9.2.2 Semantics of service primitive

Link_Capability_Discover.confirm(
Status,
SupportedLinkEventList,
SupportedLinkCommandList
)

Parameters:

Name	Type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
SupportedLinkEventList	LINK_EVENT_LIST	List of link layer events supported by the link layer. Note, this parameter is not included if Status does not indicate “Success”.
SupportedLinkCommandList	LINK_CMD_LIST	List of link layer commands supported by the link layer. Note, this parameter is not included if Status does not indicate “Success”.

7.3.9.2.3 When generated

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

7.3.9.2.4 Effect on receipt

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

7.3.10 Link_Event_Subscribe

7.3.10.1 Link_Event_Subscribe.request

7.3.10.1.1 Function

This primitive is used by MIHF (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 will not be delivered to the subscriber.

7.3.10.1.2 Semantics of service primitive

Link_Event_Subscribe.request (
RequestedLinkEventList
)

Parameter:

Name	Type	Description
RequestedLinkEventList	LINK_EVENT_LIST	List of link layer events that the subscriber would like to receive indications for.

7.3.10.1.3 When generated

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

7.3.10.1.4 Effect on receipt

The recipient responds immediately with Link_Event_Subscribe.confirm primitive.

7.3.10.2 Link_Event_Subscribe.confirm

7.3.10.2.1 Function

This primitive returns the result of the subscription request.

7.3.10.2.2 Semantics of service primitive

Link_Event_Subscribe.confirm (
Status,
ResponseLinkEventList
)

Parameters:

Name	Type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
ResponseLinkEventList	LINK_EVENT_LIST	List of successfully subscribed link events. Note, 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 a Link_Event_Subscribe.request primitive.

7.3.10.2.4 Effect on receipt

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

7.3.11 Link_Event_Unsubscribe

7.3.11.1 Link_Event_Unsubscribe.request

7.3.11.1.1 Function

This primitive is used by the MIHF (the subscriber) to unsubscribe from a set of previously subscribed link layer events.

7.3.11.1.2 Semantics of service primitive

```
Link_Event_Unsubscribe.request    (
                                  RequestedLinkEventList
                                  )
```

Parameter:

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

7.3.11.1.3 When generated

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

7.3.11.1.4 Effect on receipt

The recipient responds immediately with Link_Event_Unsubscribe.confirm primitive.

7.3.11.2 Link_Event_Unsubscribe.confirm

7.3.11.2.1 Function

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

7.3.11.2.2 Semantics of service primitive

```
Link_Event_Unsubscribe.confirm    (
                                  Status,
                                  ResponseLinkEventList
                                  )
```

Parameters:

Name	Type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
ResponseLinkEventList	LINK_EVENT_LIST	List of successfully unsubscribed link events. Note, 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 a Link_Event_Unsubscribe.request primitive.

7.3.11.2.4 Effect on receipt

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

7.3.12 Link_Get_Parameters

7.3.12.1 Link_Get_Parameters.request

7.3.12.1.1 Function

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

7.3.12.1.2 Semantics of service primitive

```
Link_Get_Parameters.request     (
                                LinkParametersRequest,
                                LinkStatesRequest,
                                LinkDescriptorsRequest
                                )
```

Parameter:

Name	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.

7.3.12.1.3 When generated

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

7.3.12.1.4 Effect on receipt

The recipient link responds with Link_Get_Parameters.confirm primitive.

7.3.12.2 Link_Get_Parameters.confirm

7.3.12.2.1 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).

7.3.12.2.2 Semantics of service primitive

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

Parameters:

Name	Type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
LinkParametersStatusList	LIST(LINK_PARAM)	A list of measurable link parameters and their current values. Note, this parameter is not included if Status does not indicate “Success”.
LinkStatesResponse	LIST(LINK_STATES_RSP)	The current link state information. Note, this parameter is not included if Status does not indicate “Success”.
LinkDescriptorsResponse	LIST(LINK_DESC_RSP)	The descriptors of a link. Note, this parameter is not included if Status does not indicate “Success”.

7.3.12.2.3 When generated

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

7.3.12.2.4 Effect on receipt

The recipient passes the link parameter values received to the MIH Users. However, if Status does not indicate “Success”, the recipient performs appropriate error handling.

7.3.13 Link_Configure_Thresholds

7.3.13.1 Link_Configure_Thresholds.request

7.3.13.1.1 Function

This primitive is used by the MIHF to configure thresholds and/or specify the time interval between periodic reports for the Link_Parameters_Report indication. This reporting period is set by Link_Configure_Threshold (see 7.3.13).

7.3.13.1.2 Semantics of service primitive

Link_Configure_Thresholds.request (
LinkConfigureParameterList
)

Parameter:

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

7.3.13.1.3 When generated

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

7.3.13.1.4 Effect on receipt

The recipient responds immediately with Link_Configure_Thresholds.confirm primitive.

7.3.13.2 Link_Configure_Thresholds.confirm

7.3.13.2.1 Function

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

7.3.13.2.2 Semantics of service primitive

Link_Configure_Thresholds.confirm (
Status,
LinkConfigureStatusList
)

Parameters:

Name	Type	Description
Status	STATUS	Status of operation. Code 3 (Authorization Failure) is not applicable.
LinkConfigureStatusList	LIST(LINK_CFG_STATUS)	A list of Link Configure Status. Note, this parameter is not included if Status does not indicate “Success”.

7.3.13.2.3 When generated

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

7.3.13.2.4 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.

7.3.14 Link_Action

7.3.14.1 Link_Action.request

7.3.14.1.1 Function

This primitive is used by the MIHF 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 can 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 can also be specified for cases where the link layer technology under consideration supports the action.

7.3.14.1.2 Semantics of Service Primitive

```
Link_Action.request        (
                            LinkAction,
                            ExecutionDelay,
                            PoALinkAddress
                           )
```

Parameters:

Name	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 instance 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 is parameter is used when DATA_FWD_REQ action is requested.

7.3.14.1.3 When generated

The MIHF generates this primitive upon request from the MIH User to perform an action on a pre-defined link layer connection.

7.3.14.1.4 Effect on receipt

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

7.3.14.2 Link_Action.confirm

7.3.14.2.1 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.

7.3.14.2.2 Semantics of Service Primitive

```
Link_Action.confirm      (
                         Status,
                         ScanResponseSet,
                         LinkActionResult
                         )
```

Parameters:

Name	Type	Description
Status	STATUS	Status of the operation. Note: Code 3 (Authorization Failure) is not applicable.
ScanResponseSet	LIST(LINK_SCAN_RSP)	(Optional) A list of discovered links and related information. Note, this parameter is not included if Status does not indicate "Success".
LinkActionResult	LINK_AC_RESULT	Specifies whether the link action was successful. Note, this parameter is not included if Status does not indicate "Success".

7.3.14.2.3 When generated

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

7.3.14.2.4 Effect on receipt

Upon receipt of this primitive, the MIHF determines the relevant MIH command that needs to be used to provide an indication or confirmation to the MIH 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 MIH_SAP primitives

The primitives defined as part of MIH_SAP are described in the following subclauses.

7.4.1 MIH_Capability_Discover

7.4.1.1 MIH_Capability_Discover.request

7.4.1.1.1 Function

This primitive is used by an MIH User to discover the capabilities of the local MIHF or a remote MIHF. When invoking this primitive to discover the capabilities of a remote MIHF, the MIH User can optionally piggyback the capability information of its local MIHF so that the two MIHFs can mutually discover each other's capabilities with a single invocation of this primitive.

7.4.1.1.2 Semantics of service primitive

MIH_Capability_Discover.request (

DestinationIdentifier,
LinkAddressList,
SupportedMihEventList,
SupportedMihCommandList,
SupportedIsQueryTypeList,
SupportedTransportList,
MBBHandoverSupport
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies the local MIHF or a remote MIHF that will be the destination of this request.
LinkAddressList	LIST(NET_TYPE_ADDR)	(Optional) A list of network type and link address pair on the local MIHF.
SupportedMIHEventList	MIH_EVT_LIST	(Optional) List of supported events on the local MIHF.
SupportedMIHCommandList	MIH_CMD_LIST	(Optional) List of supported commands on the local MIHF.
SupportedISQueryTypeList	MIH_IQ_TYPE_LST	(Optional) List of supported MIIS query types on the local MIHF.
SupportedTransportList	MIH_TRANS_LST	(Optional) List of supported transport types on the local MIHF.
MBBHandoverSupport	LIST(MBB_HO_SUPP)	(Optional) This is used to indicate if a make before break handover is supported on the local MIHF. Break before make handover is always supported.

7.4.1.1.3 When generated

This primitive is generated by an MIH User to discover the capabilities of the local MIHF or a remote MIHF. In the case of remote discovery, this primitive contains the SupportedMihEventList, SupportedMihCommandList, SupportedIsQueryTypeList, SupportedTransportList, and MBBHandoverSupport parameters of the local MIHF 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 MIHF itself, the local MIHF responds with MIH_Capability_Discover.confirm. If the destination of the request is a remote MIHF, the local MIHF shall generate a corresponding MIH_Capability_Discover request message to the remote MIHF if it does not have the capability information of the remote MIHF.

7.4.1.2 MIH_Capability_Discover.indication

7.4.1.2.1 Function

This primitive is used by an MIHF to notify an MIH User on the receipt of an MIH_Capability_Discover request message from a peer MIHF.

7.4.1.2.2 Semantics of Service primitive

MIH_Capability_Discover.indication (
SourceIdentifier,
LinkAddressList,
SupportedMihEventList,
SupportedMihCommandList,
SupportedIsQueryTypeList,
SupportedTransportList,
MBBHandoverSupport
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which is a remote MIHF.
LinkAddressList	LIST(NET_TYPE_ADDR)	(Optional) A list of network type and link address pair on the remote MIHF.
SupportedMIHEventList	MIH_EVT_LIST	(Optional) List of supported events on the remote MIHF.
SupportedMIHCommandList	MIH_CMD_LIST	(Optional) List of supported commands on the remote MIHF.
SupportedISQueryTypeList	MIH_IQ_TYPE_LST	(Optional) List of supported MIIS query types on the remote MIHF.
SupportedTransportList	MIH_TRANS_LST	(Optional) List of supported transport types on the remote MIHF.
MBBHandoverSupport	LIST(MBB_HO_SUPP)	(Optional) This is used to indicate if a make before break handover is supported on the remote MIHF. Break before make handover is always supported.

7.4.1.2.3 When generated

This primitive is used by an MIHF to notify an MIH User when an MIH_Capability_Discover request message is received. This primitive is optional since the MIHF can immediately return an MIH_Capability_Discover response message without generating this primitive to the MIH User.

7.4.1.2.4 Effect on receipt

The MIH User responds with an MIH_Capability_Discover.response primitive when an indication is received.

7.4.1.3 MIH_Capability_Discover.response

7.4.1.3.1 Function

This primitive is used by an MIH User to convey the locally supported MIH capabilities to the MIH User that invoked the MIH_Capability_Discover request.

7.4.1.3.2 Semantics of Service primitive

MIH_Capability_Discover.response(
DestinationIdentifier,
Status,
LinkAddressList,
SupportedMihEventList,
SupportedMihCommandList,
SupportedIsQueryTypeList,
SupportedTransportList,
MBBHandoverSupport
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies the remote MIHF that will 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 MIHF.
SupportedMIHEventList	MIH_EVT_LIST	(Optional) List of supported events on local MIHF.
SupportedMIHCommandList	MIH_CMD_LIST	(Optional) List of supported commands on local MIHF.
SupportedISQueryTypeList	MIH_IQ_TYPE_LST	(Optional) List of supported MIIS query types on local MIHF.
SupportedTransportList	MIH_TRANS_LST	(Optional) List of supported transport types on local MIHF.
MBBHandoverSupport	LIST(MBB_HO_SUPP)	(Optional) This is used to indicate if a make before break handover is supported on local MIHF. Break before make handover is always supported.

7.4.1.3.3 When generated

This primitive is generated by an MIH User as a response to a received MIH_Capability_Discover.indication primitive.

7.4.1.3.4 Effect on receipt

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

7.4.1.4 MIH_Capability_Discover.confirm

7.4.1.4.1 Function

This primitive is used by the MIHF to convey the supported MIH capabilities about Event Service, Command Service, and Information Service to the MIH User that invoked the MIH_Capability_Discover.request.

7.4.1.4.2 Semantics of service primitive

```
MIH_Capability_Discover.confirm (
                                SourceIdentifier,
                                Status,
                                LinkAddressList,
                                SupportedMihEventList,
                                SupportedMihCommandList,
                                SupportedIsQueryTypeList,
                                SupportedTransportList,
                                MBBHandoverSupport
                                )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
Status	STATUS	Status of operation.
LinkAddressList	LIST(NET_TYPE_ADDR)	(Optional) A list of network type and link address pair on the MIHF identified by Source Identifier.
SupportedMIHEventList	MIH_EVT_LIST	(Optional) List of supported events on the MIHF identified by Source Identifier.
SupportedMIHCommandList	MIH_CMD_LIST	(Optional) List of supported commands on the MIHF identified by Source Identifier.
SupportedISQueryTypeList	MIH_IQ_TYPE_LST	(Optional) List of supported MIIS query types on the MIHF identified by Source Identifier.
SupportedTransportList	MIH_TRANS_LST	(Optional) List of supported transport types on the MIHF identified by Source Identifier.
MBBHandoverSupport	LIST(MBB_HO_SUPP)	(Optional) This is used to indicate if a make before break handover is supported on the MIHF identified by Source Identifier. Break before make handover is always supported.

7.4.1.4.3 When generated

This primitive is invoked by a local MIHF to convey the results of a previous MIH_Capability_Discover.request primitive from an MIH User.

7.4.1.4.4 Effect on receipt

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

7.4.2 MIH_Register

7.4.2.1 MIH_Register.request

7.4.2.1.1 Function

This primitive is used by an MIH User to register the local MIHF with remote MIHF.

7.4.2.1.2 Semantics of service primitive

```
MIH_Register.request    (
                        DestinationIdentifier,
                        LinkIdentifierList,
                        RequestCode
                        )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this request.
LinkIdentifierList	LIST(LINK_ID)	List of local link identifiers.
RequestCode	REG_REQUEST_CODE	Registration request code. Depending on the request code, the MIH User can choose to either register or re-register with the remote MIHF.

7.4.2.1.3 When generated

This primitive is invoked by the MIH User when it needs to register the local MIHF with a remote MIHF.

7.4.2.1.4 Effect on receipt

On receipt, the local MIHF sends an MIH_Register request message to the destination MIHF.

7.4.2.2 MIH_Register.indication

7.4.2.2.1 Function

This primitive is used by an MIHF to notify an MIH User that an MIH_Register request message has been received.

7.4.2.2.2 Semantics of service primitive

```
MIH_Register.indication (
                        SourceIdentifier,
                        LinkIdentifierList,
                        RequestCode
                        )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which is a remote MIHF.
LinkIdentifierList	LIST(LINK_ID)	List of link identifiers of the remote MIHF.
RequestCode	REG_REQUEST_CODE	Registration request code. Depending on the request code, the MIH User can choose to either register or re-register with the remote MIHF.

7.4.2.2.3 When generated

This primitive is generated by the remote MIHF when an MIH_Register request message is received.

7.4.2.2.4 Effect on receipt

The remote MIH User will perform necessary actions to process the registration request and respond with an MIH_Register.response.

7.4.2.3 MIH_Register.response

7.4.2.3.1 Function

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

7.4.2.3.2 Semantics of service primitive

```
MIH_Register.response    (
                         DestinationIdentifier,
                         Status,
                         ValidTimeInterval
                         )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF, which will be the destination of this response.
Status	STATUS	Status of operation.
ValidTimeInterval	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. Note, this parameter is not included if Status does not indicate “Success”.

7.4.2.3.3 When generated

This primitive is invoked by the MIH 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 MIHF sends an MIH_Register response message to the destination MIHF.

7.4.2.4 MIH_Register.confirm

7.4.2.4.1 Function

This primitive is used by the local MIHF to convey the result of a registration request to an MIH User.

7.4.2.4.2 Semantics of service primitive

```
MIH_Register.confirm     (
                         SourceIdentifier,
                         Status,
                         ValidTimeInterval
                         )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which is a remote MIHF.
Status	STATUS	Status of operation.
ValidTimeInterval	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. Note, this parameter is not included if Status does not indicate “Success”.

7.4.2.4.3 When generated

This primitive is used by an MIHF to notify an MIH User the result of an MIH registration request.

7.4.2.4.4 Effect on receipt

Upon receipt, the MIH User can determine the result of the registration request.

7.4.3 MIH_DeRegister

7.4.3.1 MIH_DeRegister.request

7.4.3.1.1 Function

This primitive is used by an MIH User to deregister the local MIHF with peer MIHF.

7.4.3.1.2 Semantics of service primitive

```
MIH_DeRegister.request   (
                         DestinationIdentifier
                         )
```

Parameter:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this request.

7.4.3.1.3 When generated

This primitive is invoked by the MIH User when it needs to terminate an existing MIH registration with a remote MIHF.

7.4.3.1.4 Effect on receipt

Upon receipt, the local MIHF generates and sends an MIH_DeRegister request message to the destination MIHF.

7.4.3.2 MIH_DeRegister.indication

7.4.3.2.1 Function

This primitive is used by an MIHF to notify an MIH User that an MIH_DeRegister request message has been received.

7.4.3.2.2 Semantics of service primitive

```
MIH_DeRegister.indication(
                         SourceIdentifier
                         )
```

Parameter:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which is a remote MIHF.

7.4.3.2.3 When generated

This primitive is generated by an MIHF when an MIH_DeRegister request message is received.

7.4.3.2.4 Effect on receipt

The MIH User will perform necessary actions to process the deregistration request and respond with an MIH_DeRegister.response.

7.4.3.3 MIH_DeRegister.response

7.4.3.3.1 Function

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

7.4.3.3.2 Semantics of service primitive

```
MIH_DeRegister.response (
                         DestinationIdentifier,
                         Status
                         )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF, which will be the destination of this response.
Status	STATUS	Status of operation. Note: Code 2 (Reject) is not used.

7.4.3.3.3 When generated

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

7.4.3.3.4 Effect on receipt

Upon receipt, the local MIHF sends an MIH_DeRegister response message to the destination MIHF.

7.4.3.4 MIH_DeRegister.confirm

7.4.3.4.1 Function

This primitive is used by the local MIHF to convey the result of a deregistration request to the local MIH User.

7.4.3.4.2 Semantics of service primitive

```
MIH_DeRegister.confirm  (
                         SourceIdentifier,
                         Status
                         )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which is a remote MIHF.
Status	STATUS	Status of operation. Code 2 (Rejected) is not used.

7.4.3.4.3 When generated

This primitive is used by an MIHF to notify the local MIH User the status of MIH deregistration request.

7.4.3.4.4 Effect on receipt

Upon receipt, the MIH User can determine the status of the deregistration request.

7.4.4 MIH_Event_Subscribe

7.4.4.1 MIH_Event_Subscribe.request

7.4.4.1.1 Function

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

7.4.4.1.2 Semantics of service primitive

```
MIH_Event_Subscribe.request     (
                                DestinationIdentifier,
                                LinkIdentifier,
                                RequestedMihEventList,
                                EventConfigurationInfoList
                                )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies the local MIHF or a remote MIHF that will be the destination of this request.
LinkIdentifier	LINK_TUPLE_ID	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.
RequestedMIHEventList	MIH_EVT_LIST	List of MIH 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 MIH User when it wants to receive indications on a set of specific MIH events from the local MIHF or a remote MIHF.

7.4.4.1.4 Effect on receipt

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

7.4.4.2 MIH_Event_Subscribe.confirm

7.4.4.2.1 Function

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

7.4.4.2.2 Semantics of service primitive

MIH_Event_Subscribe.confirm (
SourceIdentifier,
Status,
LinkIdentifier,
ResponseMihEventList
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
Status	STATUS	Status of operation.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link for event subscription.
ResponseMIHEventList	MIH_EVT_LIST	List of successfully subscribed MIH events. Note, 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 MIHF at the completion of processing an MIH_Event_Subscribe.request primitive from a local MIH User or in response to the receiving of an MIH_Event_Subscribe response message from a peer MIHF.

7.4.4.2.4 Effect on receipt

The recipient MIH 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 MIH_Event_Unsubscribe

7.4.5.1 MIH_Event_Unsubscribe.request

7.4.5.1.1 Function

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

7.4.5.1.2 Semantics of service primitive

MIH_Event_Unsubscribe.request (
DestinationIdentifier,
LinkIdentifier,

RequestedMihEventList
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies the local MIHF or a remote MIHF, which will be the destination of this request.
LinkIdentifier	LINK_TUPLE_ID	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.
RequestedMIHEventList	MIH_EVT_LIST	List of MIH 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 MIH User (subscriber) that is seeking to unsubscribe from an already subscribed set of events from the local MIHF or a remote MIHF.

7.4.5.1.4 Effect on receipt

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

7.4.5.2 MIH_Event_Unsubscribe.confirm

7.4.5.2.1 Function

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

7.4.5.2.2 Semantics of service primitive

MIH_Event_Unsubscribe.confirm (
SourceIdentifier,
Status,
LinkIdentifier,
ResponseMihEventList
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
Status	STATUS	Status of operation.

LinkIdentifier	LINK_TUPLE_ID	Identifier of the link for event unsubscription.
ResponseMIHEventList	MIH_EVT_LIST	List of successfully unsubscribed link events. Note, 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 MIHF at the completion of processing an MIH_Event_Unsubscribe.request primitive from a local MIH User or in response to the receiving of an MIH_Event_Unsubscribe response message from a peer MIHF.

7.4.5.2.4 Effect on receipt

The recipient MIH User can 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 MIH_Link_Detected

7.4.6.1 MIH_Link_Detected.indication

7.4.6.1.1 Function

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

7.4.6.1.2 Semantics of the service primitive

```
MIH_Link_Detected.indication (
                    SourceIdentifier,
                    LinkDetectedInfoList
                    )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
LinkDetectedInfoList	LIST(LINK_DET_INFO)	List of link detection information.

7.4.6.1.3 When generated

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

7.4.6.1.4 Effect on receipt

MIH User dependant.

7.4.7 MIH_Link_Up

7.4.7.1 MIH_Link_Up.indication

7.4.7.1.1 Function

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

7.4.7.1.2 Semantics of the service primitive

```
MIH_Link_Up.indication  (
                        SourceIdentifier,
                        LinkIdentifier,
                        OldAccessRouter,
                        NewAccessRouter,
                        IPRenewalFlag,
                        Mobility Management Support
                        )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
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.
MobilityManagementSupport	IP_MOB_MGMT	(Optional) Indicates the type of Mobility Management Protocol supported by the new PoA.

7.4.7.1.3 When generated

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

7.4.7.1.4 Effect on receipt

MIH User dependant.

7.4.8 MIH_Link_Down

7.4.8.1 MIH_Link_Down.indication

7.4.8.1.1 Function

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

7.4.8.1.2 Semantics of the service primitive

```
MIH_Link_Down.indication (
                    SourceIdentifier,
                    LinkIdentifier,
                    OldAccessRouter,
                    ReasonCode
                    )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
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.

7.4.8.1.3 When generated

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

7.4.8.1.4 Effect on receipt

MIH User dependant.

7.4.9 MIH_Link_Parameters_Report

7.4.9.1 MIH_Link_Parameters_Report.indication

7.4.9.1.1 Function

MIH_Link_Parameters_Report indication is sent by the local MIHF to a local MIH User to report the status of a set of parameters of a local or remote link. This MIH event is either local or remote.

7.4.9.1.2 Semantics of service primitive

MIH_Link_Parameters_Report.indication (
SourceIdentifier,
LinkIdentifier,
LinkParameterReportList
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event
LinkParameterReportList	LIST(LINK_PARAM_RPT)	A list of Link Parameter Reports.

7.4.9.1.3 When generated

This notification is generated by the local MIHF 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 MIHF most likely will first receive a Link_Parameters_Report.indication from the local link layer; or
- when an MIH_Link_Parameters_Report indication message is received from a remote MIHF.

7.4.9.1.4 Effect on receipt

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

7.4.10 MIH_Link_Going_Down

7.4.10.1 MIH_Link_Going_Down.indication

7.4.10.1.1 Function

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

7.4.10.1.2 Semantics of the service primitive

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

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
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.

7.4.10.1.3 When generated

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

7.4.10.1.4 Effect on receipt

MIH User dependant.

7.4.11 MIH_Link_Handover_Imminent

7.4.11.1 MIH_Link_Handover_Imminent.indication

MIH_Link_Handover_Imminent is issued by the MIHF to report that an intra-technology link switch is about to occur. This indication directly corresponds to the link layer event Link_Handover_Imminent.indication defined in 7.3.6.

7.4.11.1.1 Function

This primitive is issued by the MIHF to report the imminent occurrence of an intra-technology link handover. This MIH event is either local or remote.

7.4.11.1.2 Semantics of service primitive

```
MIH_Link_Handover_Imminent.indication (
                                        SourceIdentifier,
                                        Old Link Identifier,
                                        New Link Identifier,
                                        OldAccessRouter,
                                        NewAccessRouter
                                        )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
OldLinkIdentifier	LINK_TUPLE_ID	Identifier of the old link.
NewLinkIdentifier	LINK_TUPLE_ID	Identifier of the new link.
OldAccessRouter	LINK_ADDR	(Optional) Link address of old Access Router.
NewAccessRouter	LINK_ADDR	(Optional) Link address of new Access Router.

7.4.11.1.3 When generated

This notification is generated by the MIHF when a link layer intra-technology handover is about to occur. The event could be triggered by the reception of a Link_Handover_Imminent.indication from a link.

7.4.11.1.4 Effect on receipt

Upper layer entities take different actions upon notification.

7.4.12 MIH_Link_Handover_Complete

7.4.12.1 MIH_Link_Handover_Complete.indication

MIH_Link_Handover_Complete indication is issued by the MIHF to report the completion of an intra-technology link handover. MIH_Link_Handover_Complete indication is a result of a Link_Handover_Complete indication from the link layer.

7.4.12.1.1 Function

This primitive is issued by the MIHF to report the completion of an intra-technology link handover. This MIH event is either local or remote.

7.4.12.1.2 Semantics of service primitive

```
MIH_Link_Handover_Complete.indication (
                                        SourceIdentifier,
                                        OldLinkIdentifier,
                                        NewLinkIdentifier,
                                        OldAccessRouter,
                                        NewAccessRouter,
                                        LinkHandoverStatus
                                        )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
OldLinkIdentifier	LINK_TUPLE_ID	Identifier of the old link
NewLinkIdentifier	LINK_TUPLE_ID	Identifier of the new link
OldAccessRouter	LINK_ADDR	(Optional) Link address of old Access Router
NewAccessRouter	LINK_ADDR	(Optional) Link address of new Access Router
LinkHandoverStatus	STATUS	Status of the link handover.

7.4.12.1.3 When generated

This notification is generated by the MIHF when a link layer intra-technology handover is completed. The event could be triggered by the reception of a Link_Handover_Complete.indication from a link.

7.4.12.1.4 Effect on receipt

Upper layer entities take different actions on this notification. An MIH User makes use of this notification to configure other layers (IP, Mobile IP) for various upper layer handovers that are needed. Transport layers (e.g., TCP) also make use of this primitive to fine tune their flow control and flow congestion mechanisms.

7.4.13 MIH_Link_PDU_Transmit_Status

7.4.13.1 MIH_Link_PDU_Transmit_Status.indication

7.4.13.1.1 Function

The MIH_Link_PDU_Transmit_Status.indication is sent to local MIHF users to notify them of a local event.

7.4.13.1.2 Semantics of the service primitive

```
MIH_Link_PDU_Transmit_Status.indication(
                    SourceIdentifier,
                    LinkIdentifier,
                    PacketIdentifier,
                    TransmissionStatus
                    )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the local MIHF where this event occurred.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link associated with the event

Name	Type	Description
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

7.4.13.1.3 When generated

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

7.4.13.1.4 Effect on receipt

MIH User dependant.

7.4.14 MIH_Link_Get_Parameters

7.4.14.1 General

An MIH_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 MIH_Link_Get_Parameters command is local or remote. For example, an MIH_Link_Get_Parameters request issued by a local upper layer helps the policy function that resides out of the MIH to make optimal handover decisions for different applications when multiple links are available in an MN. However, a remotely initiated MIH_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.

7.4.14.2 MIH_Link_Get_Parameters.request

7.4.14.2.1 Function

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

7.4.14.2.2 Semantics of the service primitive

```
MIH_Link_Get_Parameters.request (
                    DestinationIdentifier,
                    DeviceStatesRequest,
                    LinkIdentifierList,
                    GetStatusRequestSet
                    )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies the local MIHF or a remote MIHF that will be the destination of this request.

DeviceStatesRequest	DEV_STATES_REQ	(Optional) List of device states being requested.
LinkIdentifierList	LIST(LINK_ID)	List of link identifiers for which status is requested. If the list is empty, return the status of all available links.
GetStatusRequestSet	LINK_STATUS_REQ	Indicate which link status(es) is being requested.

7.4.14.2.3 When generated

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

7.4.14.2.4 Effect of receipt

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

7.4.14.3 MIH_Link_Get_Parameters.confirm

7.4.14.3.1 Function

This primitive is issued by an MIHF to report the requested status of a set of specific local or remote links in response to an MIH_Link_Get_Parameters request from a local or remote MIH User.

7.4.14.3.2 Semantics of the service primitive

```
MIH_Link_Get_Parameters.confirm (
                SourceIdentifier,
                Status,
                DeviceStatesResponseList,
                GetStatusResponseList
                )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
Status	STATUS	Status of operation.
DeviceStatesResponseList	LIST(DEV_STATES_RSP)	(Optional) List of device states responses. Note, this parameter is not included if Status does not indicate “Success”.
GetStatusResponseList	LIST(SEQUENCE(LINK_ID, LINK_STATUS_RSP))	List of link status responses. Note, this parameter is not included if Status does not indicate “Success”.

7.4.14.3.3 When generated

This primitive returns the results of an MIH_Link_Get_Parameters request to the requesting MIH User.

7.4.14.3.4 Effect of receipt

Upon receipt of the link status information, the MIH User makes appropriate decisions and takes suitable actions. However, if Status does not indicate “Success”, the recipient performs appropriate error handling.

7.4.15 MIH_Link_Configure_Thresholds

7.4.15.1 General

The MIH_Link_Configure_Thresholds is issued by an upper layer entity to configure parameter report thresholds of a lower layer. The destination of an MIH_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 MIH_Link_Parameters_Report notification shall be sent to all MIH Users that are subscriber to this threshold-crossing event.

7.4.15.2 MIH_Link_Configure_Thresholds.request

7.4.15.2.1 Function

This primitive is issued by an MIH User to configure thresholds of a lower layer link.

7.4.15.2.2 Semantics of the service primitive

```
MIH_Link_Configure_Thresholds.request (
                    DestinationIdentifier,
                    LinkIdentifier,
                    ConfigureRequestList
                    )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies the local MIHF or a remote MIHF that will be the destination of this request.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link to be configured.
ConfigureRequestList	LIST(LINK_CFG_PARAM)	A list of link threshold parameters.

7.4.15.2.3 When generated

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

7.4.15.2.4 Effect of receipt

If the destination of the request is the local MIHF itself, the local MIHF 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 MIHF, the local MIHF generates and sends an MIH_Link_Configure_Thresholds request message to the remote MIHF. Upon the receipt of the message, the remote MIHF 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.

7.4.15.3 MIH_Link_Configure_Thresholds.confirm

7.4.15.3.1 Function

This primitive is issued by an MIHF to report the result of an MIH_Link_Configure_Thresholds request.

7.4.15.3.2 Semantics of the service primitive

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

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF.
Status	STATUS	Status of operation.
LinkIdentifier	LINK_TUPLE_ID	Identifier of the link configured.
ConfigureResponseList	LIST(LINK_CFG_STATUS)	A list of the configuration status for each requested link threshold parameter. Note, this parameter is not included if Status does not indicate “Success”.

7.4.15.3.3 When generated

This primitive returns the result of an MIH_Link_Configure_Thresholds request to the requesting MIH User.

7.4.15.3.4 Effect of receipt

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

7.4.16 MIH_Link_Actions

7.4.16.1 MIH_Link_Actions.request

7.4.16.1.1 Function

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

7.4.16.1.2 Semantics of Service Primitive

The parameters of the service primitive are as follows:

MIH_Link_Actions.request (
Destination Identifier,
LinkActionsList
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies the local MIHF or a remote MIHF that will be destination of this request.
LinkActionsList	LIST(LINK_ACTION_REQ)	Specifies the suggested actions.

7.4.16.1.3 When generated

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

7.4.16.1.4 Effect on receipt

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

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

7.4.16.2 MIH_Link_Actions.confirm

7.4.16.2.1 Function

This primitive is issued by an MIHF to report the result of an MIH_Link_Actions request.

7.4.16.2.2 Semantics of the service primitive

The parameters of the primitive are as follows:

MIH_Link_Actions.confirm (
SourceIdentifier,
Status,
LinkActionsResultList
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be either the local MIHF or a remote MIHF

Status	STATUS	Status of operation.
LinkActionsResultList	LIST(LINK_ACTION_RSP)	Contain the result of the request link actions. Note, this parameter is not included if Status does not indicate “Success”.

7.4.16.2.3 When generated

This primitive returns the result of an MIH_Link_Actions.request to the requesting MIH User.

7.4.16.2.4 Effect on receipt

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

7.4.17 MIH_Net_HO_Candidate_Query

7.4.17.1 General

For network initiated handovers, the network controller provides a list of candidate network choices to the MN (via MIH_Net_HO_Candidate_Query request message). The MN indicates resources required on each of these candidate networks in the MIH_Net_HO_Candidate_Query response message. The network controller then queries each of the candidate networks for available resources (using MIH_N2N_HO_Query_Resources primitive). Once the target network has been selected, the network controller sends an MIH_Net_HO_Commit message. An example of this operation is illustrated in L.2.

7.4.17.2 MIH_Net_HO_Candidate_Query.request

7.4.17.2.1 Function

The primitive is invoked by an MIH User on a network node to communicate to a peer MIH User about its intent of handover initiation.

7.4.17.2.2 Semantics of service primitive

```
MIH_Net_HO_Candidate_Query.request (
                    DestinationIdentifier,
                    SuggestedNewLinkList,
                    QueryResourceReportFlag
                    )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this request.

SuggestedNewLinkList	LIST(LINK_POA_LIST)	A list of PoAs for each link, suggesting the new access networks to which handover initiation should be considered. The access networks towards the top of the list are more preferable than those towards the bottom of the list.
QueryResourceReportFlag	BOOLEAN	Flag to specify if resources need to be reported by MN: TRUE - Required to report resource list FALSE - Not required to report resource list.

7.4.17.2.3 When generated

This primitive is invoked by an MIH User to communicate with a remote MIH User about its intent of handover initiation. Serving PoS requests MN to provide information about resources required to initiate a handover by setting the QueryResourceReportFlag parameter.

7.4.17.2.4 Effect on receipt

Upon receipt of this primitive, the local MIHF generates and sends an MIH_Net_HO_Candidate_Query request message to the remote MIHF identified by the Destination Identifier. The remote MIHF forwards the request as an indication to the MIH User.

7.4.17.3 MIH_Net_HO_Candidate_Query.indication

7.4.17.3.1 Function

This primitive is used by an MIHF to indicate to an MIH User that an MIH_Net_HO_Candidate_Query request message was received from a remote MIHF.

7.4.17.3.2 Semantics of service primitive

```
MIH_Net_HO_Candidate_Query.indication (
                        SourceIdentifier,
                        SuggestedNewLinkList,
                        QueryResourceReportFlag
                        )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which is a remote MIHF.
SuggestedNewLinkList	LIST(LINK_POA_LIST)	A list of PoAs for each link, suggesting the new access networks to which handover initiation should be considered. The access networks towards the top of the list are more preferable than those towards the bottom of the list.
QueryResourceReportFlag	BOOLEAN	Flag to specify if resources need to be reported by MN: TRUE - Required to report resource list FALSE - Not required to report resource list.

7.4.17.3.3 When generated

This primitive is generated by an MIHF on receiving an MIH_Net_HO_Candidate_Query request message from a peer MIHF.

7.4.17.3.4 Effect on receipt

An MIH User receiving this indication shall invoke an MIH_Net_HO_Candidate_Query.response primitive towards the remote MIHF indicated by the Source Identifier in the request message.

7.4.17.4 MIH_Net_HO_Candidate_Query.response

7.4.17.4.1 Function

This primitive is used by the MIHF on an MN to respond to an MIH_Net_HO_Candidate_Query request message from a remote MIHF in the network.

7.4.17.4.2 Semantics of service primitive

```
MIH_Net_HO_Candidate_Query.response (
                                        DestinationIdentifier,
                                        Status,
                                        SourceLinkIdentifier,
                                        HandoverStatus,
                                        PreferredLinkList
                                        )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this response.
Status	STATUS	Status of operation.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the current link.
HandoverStatus	HO_STATUS	Lists the acceptance status (permit/decline) of the handover request. Note, this parameter is not included if Status does not indicate “Success”.
PreferredLinkList	LIST(RQ_RESULT)	A list of RQ_RESULT suggesting new access networks to which handover initiation should be considered. This can be different than the networks that were suggested in the handover request. The list is sorted from most preferred first to least preferred last. Note, this parameter is not included if Status does not indicate “Success” or Handover Status indicates “decline”.

7.4.17.4.3 When generated

The remote MIH User invokes this primitive in response to an MIH_Net_HO_Candidate_Query.indication from its MIHF.

7.4.17.4.4 Effect on receipt

The MIHF sends an MIH_Net_HO_Candidate_Query response message to the peer MIHF as indicated in the Destination Identifier.

7.4.17.5 MIH_Net_HO_Candidate_Query.confirm

7.4.17.5.1 Function

This primitive is used by the MIHF to confirm that an MIH_Net_HO_Candidate_Query response message was received from a peer MIHF.

7.4.17.5.2 Semantics of service primitive

MIH_Net_HO_Candidate_Query.confirm (
SourceIdentifier,
Status,
SourceLinkIdentifier,
HandoverStatus,
PreferredLinkList
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Contains the MIHF ID of the MN that sent the MIH_Net_HO_Candidate_Query response message.
Status	STATUS	Status of operation.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the current link.
HandoverStatus	HO_STATUS	Lists the acceptance status (permit/decline) of the handover request. Note, this parameter is not included if Status does not indicate “Success”.
PreferredLinkList	LIST(RQ_RESULT)	A list of RQ_RESULT suggesting new access networks to which handover initiation should be considered. This can be different than the networks that were suggested in the handover request. The list is sorted from most preferred first to least preferred last. Note, this parameter is not included if Status does not indicate “Success” or Handover Status indicates “decline”.

7.4.17.5.3 When generated

This primitive is generated by the MIHF on receiving an MIH_Net_HO_Candidate_Query response message from a peer MIHF.

7.4.17.5.4 Effect on receipt

On receiving the primitive the entity that originally initiated the handover request decides to carry out the handover or abort it based on the primitive. However, if Status does not indicate “Success”, the recipient ignores any other returned values and, instead, performs appropriate error handling.

7.4.18 MIH_MN_HO_Candidate_Query

7.4.18.1 MIH_MN_HO_Candidate_Query.request

7.4.18.1.1 Function

This primitive is used by MIH Users on an MN to inform MIHF to query candidates for possible handover initiation. The request includes queries on QoS resources and/or whether IP address configuration method of the ongoing data sessions can be supported in the candidate network. This primitive also includes the current IP configuration server address (e.g., DHCP server, foreign agent (FA) IP address, AR IP address) when the current IP configuration method is included.

7.4.18.1.2 Semantics of service primitive

```
MIH_MN_HO_Candidate_Query.request (
                    DestinationIdentifier,
                    SourceLinkIdentifier,
                    CandidateLinkList,
                    QoSResourceRequirements,
                    IPConfigurationMethods,
                    DHCPServerAddress,
                    FAAddress,
                    AccessRouterAddress
                    )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this request.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the source link for handover.
CandidateLinkList	LIST(LINK_POA_LIST)	A list of PoAs, identifying candidate networks to which handover needs to be initiated. The list is sorted from most preferred first to least preferred last.
QoSResourceRequirements	QOS_LIST	Minimal QoS resources required at the candidate network.
IPConfigurationMethods	IP_CFG_MTHDS	(Optional) Current IP configuration methods.
DHCPServerAddress	IP_ADDR	(Optional) IP address of current DHCP Server. It is only included when MN is using dynamic address configuration
FAAddress	IP_ADDR	(Optional) IP address of current Foreign Agent. It is only included when MN is using Mobile IPv4
AccessRouterAddress	IP_ADDR	(Optional) IP address of current Access Router. It is only included when MN is using IPv6

7.4.18.1.3 When generated

This primitive is generated by an MIH User in the MN that wants to query other candidate networks for a possible handover. MN uses the QueryResourceList parameter to notify the serving PoS of the minimal resource requirement at the candidate networks in order for the handover to be successful. An MIH User on

MN generates this primitive when it wants to query IP address related information from the candidate networks before handover.

7.4.18.1.4 Effect on receipt

Upon receipt of this primitive, the local MIHF generates and sends an MIH_MN_HO_Candidate_Query request message to the remote MIHF identified by the Destination Identifier.

7.4.18.2 MIH_MN_HO_Candidate_Query.indication

7.4.18.2.1 Function

This primitive is used by MIHF to indicate the receipt of MIH_MN_HO_Candidate_Query request message from an MN.

7.4.18.2.2 Semantics of service primitive

```
MIH_MN_HO_Candidate_Query.indication (
                    SourceIdentifier,
                    SourceLinkIdentifier,
                    CandidateLinkList,
                    QoSResourceRequirements,
                    IPConfigurationMethods,
                    DHCPServerAddress,
                    FAAddress,
                    AccessRouterAddress
                    )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which is a remote MIHF.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the source link for handover.
CandidateLinkList	LIST(LINK_POA_LIST)	A list of PoAs, identifying candidate networks to which handover needs to be initiated. The list is sorted from most preferred first to least preferred last.
QoSResourceRequirements	QOS_LIST	Minimal QoS resources required at the candidate network.
IPConfigurationMethods	IP_CFG_MTHDS	(Optional) Current IP configuration methods.
DHCPServerAddress	IP_ADDR	(Optional) IP address of current DHCP Server. It is only included when MN is using dynamic address configuration
FAAddress	IP_ADDR	(Optional) IP address of current Foreign Agent. It is only included when MN is using Mobile IPv4
AccessRouterAddress	IP_ADDR	(Optional) IP address of current Access Router. It is only included when MN is using IPv6

7.4.18.2.3 When generated

This primitive is generated by MIHF on receiving MIH_MN_HO_Candidate_Query request message from a peer MIHF in an MN.

7.4.18.2.4 Effect on receipt

The MIH User invokes MIH_N2N_HO_Query_Resources.request primitive to exchange MIH_N2N_HO_Query_Resource messages with MIHF in one or more candidate networks under consideration before invoking the MIH_MN_HO_Candidate_Query.response primitive.

7.4.18.3 MIH_MN_HO_Candidate_Query.response

7.4.18.3.1 Function

This primitive is used by MIH Users to inform MIHF of the result of the candidate query request.

7.4.18.3.2 Semantics of service primitive

```
MIH_MN_HO_Candidate_Query.response (
                DestinationIdentifier,
                Status,
                SourceLinkIdentifier,
                PreferredCandidateLinkList
                )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this response.
Status	STATUS	Status of operation.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the source link.
PreferredCandidateLinkList	LIST(RQ_RESULT)	A list of RQ_RESULT suggesting new access networks to which handover initiation should be considered. This can be different than the networks that were suggested in the handover request. The list is sorted from most preferred first to least preferred last. Note, this parameter is not included if Status does not indicate “Success” or Handover Status indicates “decline”.

7.4.18.3.3 When generated

The MIH User invokes this primitive in response to an MIH_MN_HO_Candidate_Query request message from a peer MIHF entity in MN and possibly after the exchange of MIH_N2N_HO_Query_Resources messages with the MIHF in the candidate networks.

7.4.18.3.4 Effect on receipt

Upon receipt of this primitive MIHF sends a response message to the destination.

7.4.18.4 MIH_MN_HO_Candidate_Query.confirm

7.4.18.4.1 Function

This primitive is used by MIHF to inform MIH Users of the receipt of candidate query and IP address related information response.

7.4.18.4.2 Semantics of service primitive

MIH_MN_HO_Candidate_Query.confirm (
SourceIdentifier,
Status,
SourceLinkIdentifier,
PreferredCandidateLinkList
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which is a remote MIHF.
Status	STATUS	Status of operation.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the source link.
PreferredCandidateLinkList	LIST(RQ_RESULT)	A list of RQ_RESULT suggesting new access networks to which handover initiation should be considered. This can be different than the networks that were suggested in the handover request. The list is sorted from most preferred first to least preferred last. Note, this parameter is not included if Status does not indicate “Success” or Handover Status indicates “decline”.

7.4.18.4.3 When generated

This primitive is generated by MIHF on receiving MIH_MN_HO_Candidate_Query response message from a peer MIHF in the network.

7.4.18.4.4 Effect on receipt

On receiving the primitive the MIH User entity that originally initiated the candidate query request can decide to choose the candidate network for handover or abort it based on the list of PoA, available resources, and the IP address related information. However, if Status does not indicate “Success”, the recipient ignores any other returned values and, instead, performs appropriate error handling.

7.4.19 MIH_N2N_HO_Query_Resources

7.4.19.1 MIH_N2N_HO_Query_Resources.request

7.4.19.1.1 Function

This primitive is used by an MIHF on the serving network to communicate with its peer MIHF on the candidate network. This is used to query the available link resource and IP address related information of the candidate network.

7.4.19.1.2 Semantics of service primitive

```
MIH_N2N_HO_Query_Resources.request (
                    DestinationIdentifier,
                    QoSResourceRequirements,
                    IPConfigurationMethods,
                    DHCPServerAddress,
                    FAAddress,
                    AccessRouterAddress,
                    CandidateLinkList
                    )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this request.
QoSResourceRequirements	QOS_LIST	Minimal QoS resources required at the candidate network.
IPConfigurationMethods	IP_CFG_MTHDS	(Optional) Current IP configuration methods.
DHCPServerAddress	IP_ADDR	(Optional) IP address of current DHCP Server. It is only included when MN is using dynamic address configuration
FAAddress	IP_ADDR	(Optional) IP address of current Foreign Agent. It is only included when MN is using Mobile IPv4
AccessRouterAddress	IP_ADDR	(Optional) IP address of current Access Router. It is only included when MN is using IPv6
CandidateLinkList	LIST(LINK_ID)	(Optional) A list of candidate links (i.e., APs or BSs) on a specific candidate network. In this list, each link is indicated by its link type and a PoA link address.

7.4.19.1.3 When generated

In the case of mobile-initiated handover, this primitive is generated after receiving the MIH_MN_HO_Candidate_Query request message from the MIHF on the MN. In the case of network-initiated handover, this primitive is generated after receiving the MIH_Net_HO_Candidate_Query response message from the MN.

7.4.19.1.4 Effect on receipt

Upon receipt of this primitive MIHF shall send a request message to the destination.

7.4.19.2 MIH_N2N_HO_Query_Resources.indication

7.4.19.2.1 Function

The MIHF on the candidate network indicates that an MIH_N2N_HO_Query_Resources request message is received from a remote MIHF on the serving network so that the upper layer entity can identify the link resource usage and provide IP address related information for the impending handover.

7.4.19.2.2 Semantics of service primitive

MIH_N2N_HO_Query_Resources.indication (
SourceIdentifier,
QoSResourceRequirements,
IPConfigurationMethods,
DHCPServerAddress,
FAAddress,
AccessRouterAddress,
CandidateLinkList
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Contains the MIHF ID of the node that sent the MIH_N2N_HO_Candidate_Query request message.
QoSResourceRequirements	QOS_LIST	Minimal QoS resources required at the candidate network.
IPConfigurationMethods	IP_CFG_MTHDS	(Optional) Current IP configuration methods.
DHCPServerAddress	IP_ADDR	(Optional) IP address of current DHCP Server. It is only included when MN is using dynamic address configuration
FAAddress	IP_ADDR	(Optional) IP address of current Foreign Agent. It is only included when MN is using Mobile IPv4
AccessRouterAddress	IP_ADDR	(Optional) IP address of current Access Router. It is only included when MN is using IPv6
CandidateLinkList	LIST(LINK_ID)	(Optional) A list of candidate links (i.e., APs or BSs) on a specific candidate network. In this list, each link is indicated by its link type and a PoA link address.

7.4.19.2.3 When generated

This primitive is generated by MIHF when the MIHF on the candidate network receives MIH_N2N_HO_Query_Resources request message from a peer MIHF on the serving network.

7.4.19.2.4 Effect on receipt

The MIH User on the candidate network identifies the link resource usage for the impending handover. It also replies with MIH_N2N_HO_Query_Resources.response primitive.

7.4.19.3 MIH_N2N_HO_Query_Resources.response

7.4.19.3.1 Function

This primitive is used by an MIHF on the candidate network to communicate with its peer MIHF on the serving network that sent out an MIH_N2N_HO_Query_Resources request message. This is used to notify the MIHF on the serving network of the link resource status of the candidate network. It is also used to provide IP address related information of the candidate networks.

7.4.19.3.2 Semantics of service primitive

MIH_N2N_HO_Query_Resources.response (
DestinationIdentifier,
Status,
ResourceStatus,
CandidateLinkList,
IPAddressInformationStatus
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_N2N_HO_Query_Resources request message.
Status	STATUS	Status of operation.
ResourceStatus	LINK_RES_STATUS	Specifies whether requested resources are available or not at the new PoA. Note, this parameter is not included if Status does not indicate "Success".
CandidateLinkList	LIST(RQ_RESULT)	A list of RQ_RESULT of the candidate links (i.e., APs or BSs). Note, this parameter is not included if Status does not indicate "Success" or Resource Status indicates "Not Available".
IPAddressInformationStatus	IP_CFG_STATUS	(Optional) The availability of IP configuration methods. Note, this parameter is not included if Status does not indicate "Success".

7.4.19.3.3 When generated

The MIHF on the candidate network invokes this primitive in response to an MIH_N2N_HO_Query_Resources request message from a peer MIHF entity on the serving network.

7.4.19.3.4 Effect on receipt

Upon receipt of this primitive MIHF sends a response message to the destination.

7.4.19.4 MIH_N2N_HO_Query_Resources.confirm

7.4.19.4.1 Function

This primitive is used by the MIHF on the serving network to respond with the result of any resource preparation for the impending handover and to notify the link resource status of the candidate network. It also carries IP address related information on the candidate networks to MIH Users on the serving network.

7.4.19.4.2 Semantics of service primitive

```
MIH_N2N_HO_Query_Resources.confirm (
                    SourceIdentifier,
                    Status,
                    ResourceStatus,
                    CandidateLinkList,
                    IPAddressInformationStatus
                    )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_N2N_HO_Candidate_Query response message.
Status	STATUS	Status of operation.
ResourceStatus	LINK_RES_STATUS	Specifies whether requested resources are available or not at the new PoA. Note, this parameter is not included if Status does not indicate “Success”.
CandidateLinkList	LIST(RQ_RESULT)	A list of RQ_RESULT of the candidate links (i.e., APs or BSs). Note, this parameter is not included if Status does not indicate “Success” or Resource Status indicates “Not Available”.
IPAddressInformation Status	IP_CFG_STATUS	(Optional) The availability of IP configuration methods. Note, this parameter is not included if Status does not indicate “Success”.

7.4.19.4.3 When generated

This primitive is generated by the MIHF when the MIHF on the serving network receives an MIH_N2N_HO_Query_Resources response message from a peer MIHF on the candidate network.

7.4.19.4.4 Effect on receipt

After receiving this primitive, the MIH User on the serving network sends an MIH_MN_HO_Candidate_Query.response primitive to the MIHF in case it was indicated with MIH_MN_HO_Candidate_Query.indication primitive before. However, if Status does not indicate “Success”, the recipient ignores any other returned values and, instead, performs appropriate error handling.

7.4.20 MIH_MN_HO_Commit

7.4.20.1 MIH_MN_HO_Commit.request

7.4.20.1.1 Function

This primitive is used by MIH Users on an MN to notify the serving network of the decided target network information.

7.4.20.1.2 Semantics of service primitive

MIH_MN_HO_Commit.request (
DestinationIdentifier,
LinkType,
TargetNetworkInfo
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This specifies the MIHF ID of the serving network that is the target of this primitive.
LinkType	LINK_TYPE	This contains the target link type.
TargetNetworkInfo	TGT_NET_INFO	This contains the target network information.

7.4.20.1.3 When generated

The MIH User generates this primitive to notify the serving network of the target network information.

7.4.20.1.4 Effect on receipt

Upon receipt of this primitive, MIHF on the mobile node sends the corresponding MIH_MN_HO_Commit request message to the serving network.

7.4.20.2 MIH_MN_HO_Commit.indication

7.4.20.2.1 Function

This primitive is generated by an MIHF on the serving network to indicate that an MIH_MN_HO_Commit request message has been received from a peer MIHF on the mobile node.

7.4.20.2.2 Semantics of service primitive

MIH_MN_HO_Commit.indication (
SourceIdentifier,
LinkType,
TargetNetworkInfo
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This specifies the MIHF ID of the mobile node that sent the MIH_MN_HO_Commit request message.
LinkType	LINK_TYPE	This contains the target link type.
TargetNetworkInfo	TGT_NET_INFO	This contains the target network information.

7.4.20.2.3 When generated

This primitive is generated by an MIHF on the serving network when receiving an MIH_MN_HO_Commit request message from a peer MIHF on the mobile node.

7.4.20.2.4 Effect on receipt

Upon receipt of this primitive MIH User on the serving network replies with an MIH_MN_HO_Commit.response primitive. MIH User may invoke the MIH_N2N_HO_Commit.request primitive to reserve the resource at the target network.

7.4.20.3 MIH_MN_HO_Commit.response

7.4.20.3.1 Function

This primitive is used by an MIH User on the serving network to communicate with a peer MIH User on the mobile node from which an MIH_MN_HO_Commit request message is received.

```
MIH_MN_HO_Commit.response (
                DestinationIdentifier,
                Status,
                LinkType,
                TargetNetworkInfo
                )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This specifies the MIHF ID of the mobile node that sent the MIH_MN_HO_Commit request message.
Status	STATUS	Status of operation.
LinkType	LINK_TYPE	This contains the target link type.
TargetNetworkInfo	TGT_NET_INFO	This contains the target network information.

7.4.20.3.2 When generated

This primitive is generated in response to an MIH_MN_HO_Commit.indication primitive.

7.4.20.3.3 Effect on receipt

When receiving this primitive from the MIH User, the MIHF on the Serving PoS sends the corresponding MIH_MN_HO_Commit response message to its peer MIHF on the mobile node.

7.4.20.4 MIH_MN_HO_Commit.confirm

7.4.20.4.1 Function

This primitive is generated by the MIHF on the mobile node to confirm that an MIH_MN_HO_Commit response message is received from a peer MIHF on the serving network.

```
MIH_MN_HO_Commit.confirm (
                    SourceIdentifier,
                    Status,
                    LinkType,
                    TargetNetworkInfo
                    )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This specifies the MIHF ID of the Serving PoS that sent the MIH_MN_HO_Commit response message.
Status	STATUS	Status of operation.
LinkType	LINK_TYPE	This contains the target link type.
TargetNetworkInfo	TGT_NET_INFO	This contains the target network information.

7.4.20.4.2 When generated

This primitive is generated by the MIHF on the mobile node when it receives an MIH_MN_HO_Commit response message from a peer MIHF on the serving network.

7.4.20.4.3 Effect on receipt

Upon receipt, the MIH User on the mobile node is informed about the status of the previously issued target notification request.

7.4.21 MIH_Net_HO_Commit

7.4.21.1 MIH_Net_HO_Commit.request

7.4.21.1.1 Function

This primitive is used by an MIH User on the network to communicate with the remote MIH User on the MN. The primitive is used to request the peer MIH User the commitment to perform a network-controlled or network-assisted link handover based on selected choices for candidate networks and PoA.

7.4.21.1.2 Semantics of service primitive

```
MIH_Net_HO_Commit.request     (
                              DestinationIdentifier,
                              LinkType,
                              TargetNetworkInfoList,
                              AssignedResourceSet,
                              LinkActionExecutionDelay,
                              LinkActionsList
                              )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	Specifies the MIHF ID of the MN MIHF that is to be committed.
LinkType	LINK_TYPE	Contains target link type.
TargetNetworkInfoList	LIST(TGT_NET_INFO)	This list contains target network information for assisting the mobile node to perform a handover.
AssignedResourceSet	ASGN_RES_SET	This includes the set of resource parameters assigned to the MN for performing the handover.
LinkActionExecutionDelay	UNSIGNED_INT(2)	Time (in ms) to elapse before an action needs to be taken. A value of 0 indicates that the action is taken immediately. Time elapsed is calculated from the instance the command arrives until the time when the execution of the action is carried out.
LinkActionsList	LIST(LINK_ACTION_REQ)	(Optional) A list of network controlled handover actions for the links.

7.4.21.1.3 When generated

The MIH User generates this primitive to order specific handover actions on one or more links.

7.4.21.1.4 Effect on receipt

Upon receipt of this primitive MIHF shall send an MIH_NET_HO_Commit request message to the destination.

7.4.21.2 MIH_Net_HO_Commit.indication

7.4.21.2.1 Function

This primitive is used by an MIHF to indicate that an MIH_Net_HO_Commit request message has been received from a peer MIHF.

7.4.21.2.2 Semantics of service primitive

```
MIH_Net_HO_Commit.indication (
                              SourceIdentifier,
                              LinkType,
                              TargetNetworkInfoList,
```

AssignedResourceSet,
LinkActionExecutionDelay,
LinkActionsList
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_Net_HO_Commit request message.
LinkType	LINK_TYPE	Contains target link type.
TargetNetworkInfoList	LIST(TGT_NET_INFO)	This list contains target network information for assisting the mobile node to perform a handover.
AssignedResourceSet	ASGN_RES_SET	This includes the set of resource parameters assigned to the MN for performing the handover.
LinkActionExecutionDelay	UNSIGNED_INT(2)	Time (in ms) to elapse before an action needs to be taken. A value of 0 indicates that the action is taken immediately. Time elapsed is calculated from the instance the command arrives until the time when the execution of the action is carried out.
LinkActionsList	LIST(LINK_ACTION_REQ)	(Optional) A list of network controlled handover actions for the links.

7.4.21.2.3 When generated

This primitive is generated by an MIHF on receiving an MIH_Net_HO_Commit request message from a peer MIHF.

7.4.21.2.4 Effect on receipt

The MIH User receiving this primitive replies with an MIH_Net_HO_Commit.response primitive. Only the applicable actions in the Link Actions List are executed. The non-applicable link actions indicate failed actions when preparing the response.

7.4.21.3 MIH_Net_HO_Commit.response

7.4.21.3.1 Function

This primitive is used by an MIHF to communicate with a peer MIHF from which an MIH_Net_HO_Commit request message is received. The primitive is used to communicate the response of a handover commit request.

7.4.21.3.2 Semantics of service primitive

MIH_Net_HO_Commit.response (
DestinationIdentifier,
Status,
LinkType,
TargetNetworkInfo,
LinkActionsResultList
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_Net_HO_Commit request message.
Status	STATUS	Status of operation.
LinkType	LINK_TYPE	Contains target link type. Note, this parameter is not included if Status does not indicate “Success”.
TargetNetworkInfo	TGT_NET_INFO	Contains target network information for handover. Note, this parameter is not included if Status does not indicate “Success”.
LinkActionsResultList	LIST(LINK_ACTION_RSP)	(Optional) A list of link actions result. This parameter is present if and only if Link Actions List parameter is present in MIH_Net_HO_Commit.indication. Note, this parameter is not included if Status does not indicate “Success”.

7.4.21.3.3 When generated

This primitive is generated in response to an MIH_Net_HO_Commit.indication primitive.

7.4.21.3.4 Effect on receipt

Upon receipt of this primitive MIHF shall send an MIH_Net_HO_Commit response message to the destination.

7.4.21.4 MIH_Net_HO_Commit.confirm

7.4.21.4.1 Function

This primitive is used by the MIHF to confirm that an MIH_Net_HO_Commit response message is received from a peer MIHF.

7.4.21.4.2 Semantics of service primitive

```
MIH_Net_HO_Commit.confirm (
                    SourceIdentifier,
                    Status,
                    LinkType,
                    TargetNetworkInfo,
                    LinkActionsResultList
                    )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_Net_HO_Commit response message.
Status	STATUS	Status of operation.
LinkType	LINK_TYPE	Contains target link type. Note, this parameter is not included if Status does not indicate “Success”.
TargetNetworkInfo	TGT_NET_INFO	Contains target network information for handover. Note, this parameter is not included if Status does not indicate “Success”.
LinkActionsResultList	LIST(LINK_ACTION_RSP)	(Optional) A list of link actions result. This parameter is present if and only if Link Actions List parameter is present in MIH_Net_HO_Commit.indication. Note, this parameter is not included if Status does not indicate “Success”.

7.4.21.4.3 When generated

This primitive is generated by the MIHF on receiving an MIH_Net_HO_Commit response message from a peer MIHF.

7.4.21.4.4 Effect on receipt

Upon receipt, the old serving PoS is informed about the status of the previously issued command request.

Since MIH_Net_HO_Commit request message contains actions to effect the handover, and the response has the status of those actions, the link to the old PoS may not be accessible (e.g. break before make) to receive the response before L3 connectivity has been established on the new link and only if the MN knows the old PoS L3 address thus, the old PoS may not receive this response.

7.4.22 MIH_N2N_HO_Commit

7.4.22.1 MIH_N2N_HO_Commit.request

7.4.22.1.1 Function

This primitive is used by an MIH User on the serving network to inform a selected target network that an MN is about to move to the target network.

7.4.22.1.2 Semantics of service primitive

The parameters of the primitive are as follows:

```
MIH_N2N_HO_Commit.request (
                    DestinationIdentifier,
                    MNIdentifier,
                    TargetMNLinkIdentifier,
```

TargetPoA,
RequestedResourceSet
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this request.
MNIdentifier	MIHF_ID	This identifies the MIHF of the MN that commits to perform handover action.
TargetMNLinkIdentifier	LINK_ID	This is the identifier of the MN's target link for which resources are requested.
TargetPoA	LINK_ADDR	This is the link address of the target Point of Attachment (AP/BS).
RequestedResourceSet	REQ_RES_SET	This includes the set of parameters required for performing MN admission control and resource reservation at the target network

7.4.22.1.3 When generated

The MIH User on the serving network invokes this primitive when a single target network has been decided.

7.4.22.1.4 Effect on receipt

Upon receipt of this primitive, the local MIHF generates and sends an MIH_N2N_HO_Commit request message to the remote MIHF on the selected target network identified by the Destination Identifier.

7.4.22.2 MIH_N2N_HO_Commit.indication

7.4.22.2.1 Function

This primitive is used by an MIHF to indicate that an MIH_N2N_HO_Commit request message has been received from a peer MIHF on serving network.

7.4.22.2.2 Semantics of service primitive

The parameters of the primitive are as follows:

MIH_N2N_HO_Commit.indication(
SourceIdentifier,
MNIdentifier,
TargetMNLinkIdentifier,
TargetPoA,
RequestedResourceSet
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	This identifies the invoker of this primitive, which can be a remote MIHF.
MNIdentifier	MIHF_ID	This identifies the MIHF of the MN that commits to perform handover action.
TargetMNLinkIdentifier	LINK_ID	This is the identifier of the MN's target link for which resources are requested.
TargetPoA	LINK_ADDR	This is the link address of the target Point of Attachment (AP/BS).
RequestedResourceSet	REQ_RES_SET	This includes the set of parameters required for performing MN admission control and resource reservation at the target network

7.4.22.2.3 When generated

This primitive is generated by an MIHF on receiving an MIH_N2N_HO_Commit request message from a peer MIHF on the serving network.

7.4.22.2.4 Effect on receipt

Upon receipt of this primitive, MIH User generates an MIH_N2N_HO_Commit.response primitive.

7.4.22.3 MIH_N2N_HO_Commit.response

7.4.22.3.1 Function

This primitive is used by an MIH User to respond to an MIH_N2N_HO_Commit.indication primitive.

7.4.22.3.2 Semantics of service primitive

The parameters of the primitive are as follows:

```
MIH_N2N_HO_Commit.response (
                        DestinationIdentifier,
                        Status,
                        MNIdentifier,
                        TargetLinkIdentifier,
                        AssignedResourceSet
                        )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this request.
Status	STATUS	Status of operation.

MNIdentifier	MIHF_ID	This identifies the MIHF of the MN that commits to perform handover action.
TargetLinkIdentifier	LINK_TUPLE_ID	This contains the identifier of the target Point of Attachment (AP/BS) for the MN. Note, this parameter is not included if Status does not indicate “Success”.
AssignedResourceSet	ASGN_RES_SET	This includes the set of resource parameters assigned by the target network to the MN. Note, this parameter is not included if Status does not indicate “Success”.

7.4.22.3.3 When generated

This primitive is generated by an MIHF User in response to a received MIH_N2N_HO_Commit.indication primitive.

7.4.22.3.4 Effect on receipt

Upon receipt, the MIHF generates and sends an MIH_N2N_HO_Commit response message to the peer MIHF on the serving network that sent an MIH_N2N_HO_Commit request message.

7.4.22.4 MIH_N2N_HO_Commit.confirm

7.4.22.4.1 Function

This primitive is used by the MIHF to confirm that an MIH_N2N_HO_Commit response message is received from a peer MIHF on the selected target network.

7.4.22.4.2 Semantics of service primitive

The parameters of the primitive are as follows:

```
MIH_N2N_HO_Commit.confirm (
                SourceIdentifier,
                Status,
                MNIdentifier,
                TargetLinkIdentifier,
                AssignedResourceSet
                )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_N2N_HO_Commit response message.
Status	STATUS	Status of operation.
MNIdentifier	MIHF_ID	This identifies the MIHF of the MN that commits to perform handover action.

TargetLinkIdentifier	LINK_TUPLE_ID	This contains the identifier of the target Point of Attachment (AP/BS) for the MN. Note, this parameter is not included if Status does not indicate "Success".
AssignedResourceSet	ASGN_RES_SET	This includes the set of resource parameters assigned by the target network to the MN. Note, this parameter is not included if Status does not indicate "Success".

7.4.22.4.3 When generated

This primitive is generated by the MIHF on receiving an MIH_N2N_HO_Commit response message from a peer MIHF on the selected target network.

7.4.22.4.4 Effect on receipt

Upon receipt, the serving network is informed about the status of the previously issued command request so that it can react accordingly. For instance, the serving network determines that the handover procedure is acknowledged by the target network and it can notify the MN to perform handover. However, if Status does not indicate "Success", the recipient ignores any other returned values and, instead, performs appropriate error handling.

7.4.23 MIH_MN_HO_Complete

7.4.23.1 MIH_MN_HO_Complete.request

7.4.23.1.1 Function

This primitive is optionally used by MIH Users to indicate the completion of MIH level handover aiding procedure.

7.4.23.1.2 Semantics of service primitive

```
MIH_MN_HO_Complete.request (
                    DestinationIdentifier,
                    SourceLinkIdentifier,
                    TargetLinkIdentifier,
                    HandoverResult
                    )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	This identifies a remote MIHF that will be the destination of this request.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the source link of the handover.
TargetLinkIdentifier	LINK_TUPLE_ID	This identifies the target link.
HandoverResult	HO_RESULT	Handover result.

7.4.23.1.3 When generated

This primitive is generated when MIH level handover procedure is complete.

7.4.23.1.4 Effect on receipt

Upon receipt of this primitive, the local MIHF generates and sends an MIH_MN_HO_Complete request message to the remote MIHF identified by the Destination Identifier.

7.4.23.2 MIH_MN_HO_Complete.indication

7.4.23.2.1 Function

This primitive is used by MIHF to inform MIH Users locally that an MIH_MN_HO_Complete request message is received.

7.4.23.2.2 Semantics of service primitive

```
MIH_MN_HO_Complete.indication (
                    SourceIdentifier,
                    SourceLinkIdentifier,
                    TargetLinkIdentifier,
                    HandoverResult
                    )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_MN_HO_Complete request message.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the source link of the handover.
TargetLinkIdentifier	LINK_TUPLE_ID	This identifies the target link.
HandoverResult	HO_RESULT	Handover result.

7.4.23.2.3 When generated

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

7.4.23.2.4 Effect on receipt

This indicates the completion of the handover. A corresponding response is generated.

7.4.23.3 MIH_MN_HO_Complete.response

7.4.23.3.1 Function

This primitive is used by MIH Users to send a response to the MIH_MN_HO_Complete request.

7.4.23.3.2 Semantics of service primitive

MIH_MN_HO_Complete.response (
DestinationIdentifier,
Status,
SourceLinkIdentifier,
TargetLinkIdentifier
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_MN_HO_Complete request message.
Status	STATUS	Status of operation.
SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the source link of the handover.
TargetLinkIdentifier	LINK_TUPLE_ID	This identifies the target link.

7.4.23.3.3 When generated

This primitive is generated when MIH_MN_HO_Complete request message is received.

7.4.23.3.4 Effect on receipt

This indicates the completion of the MIH level handover aiding procedure.

7.4.23.4 MIH_MN_HO_Complete.confirm

7.4.23.4.1 Function

This primitive is used by MIHF to inform MIH Users locally that an MIH_MN_HO_Complete response message is received.

7.4.23.4.2 Semantics of service primitive

MIH_MN_HO_Complete.confirm (
SourceIdentifier,
Status,
SourceLinkIdentifier,
TargetLinkIdentifier
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_MN_HO_Complete response message.
Status	STATUS	Status of operation.

SourceLinkIdentifier	LINK_TUPLE_ID	This identifies the source link of the handover.
TargetLinkIdentifier	LINK_TUPLE_ID	This identifies the target link.

7.4.23.4.3 When generated

MIHF generates this primitive when an MIH_MN_HO_Complete response message is received.

7.4.23.4.4 Effect on receipt

This indicates the completion of the MIH level handover aiding procedure.

7.4.24 MIH_N2N_HO_Complete

7.4.24.1 MIH_N2N_HO_Complete.request

7.4.24.1.1 Function

This primitive is used by an MIH User in the network to communicate with a peer network MIH entity about the completion of handover operation.

7.4.24.1.2 Semantics of service primitive

```
MIH_N2N_HO_Complete.request (
                DestinationIdentifier,
                MNIdentifier,
                SourceLinkIdentifier,
                TargetLinkIdentifier,
                HandoverResult
                )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	Identify the MIHF ID of the destination node.
MNIdentifier	MIHF_ID	This identifies the MIHF on MN.
SourceLinkIdentifier	LINK_TUPLE_ID	Specifies the source link of the handover.
TargetLinkIdentifier	LINK_TUPLE_ID	Specifies the target link of the handover.
HandoverResult	HO_RESULT	Handover result.

7.4.24.1.3 When generated

The MIH User invokes this primitive when handover operations have been completed.

7.4.24.1.4 Effect on receipt

Upon receipt of this primitive, the local MIHF generates and sends an MIH_N2N_HO_Complete request message to the remote MIHF identified by the Destination Identifier.

7.4.24.2 MIH_N2N_HO_Complete.indication

7.4.24.2.1 Function

This primitive is used by the MIHF to indicate the status of the handover operation.

7.4.24.2.2 Semantics of service primitive

MIH_N2N_HO_Complete.indication(
SourceIdentifier,
MNIdentifier,
SourceLinkIdentifier,
TargetLinkIdentifier,
HandoverResult
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_N2N_HO_Complete request message.
MNIdentifier	MIHF_ID	This identifies the MIHF on MN.
SourceLinkIdentifier	LINK_TUPLE_ID	Specifies the source link of the handover.
TargetLinkIdentifier	LINK_TUPLE_ID	Specifies the target link of the handover.
HandoverResult	HO_RESULT	Handover result.

7.4.24.2.3 When generated

This primitive is generated by the MIHF on receiving an MIH_N2N_HO_Complete request message from a peer MIHF.

7.4.24.2.4 Effect on receipt

The MIH User receiving this primitive replies with an MIH_N2N_HO_Complete.response primitive.

7.4.24.3 MIH_N2N_HO_Complete.response

7.4.24.3.1 Function

This primitive is used to send a response to a handover complete request.

7.4.24.3.2 Semantics of service primitive

MIH_N2N_HO_Complete.response (
DestinationIdentifier,
Status,
MNIdentifier,
SourceLinkIdentifier,
TargetLinkIdentifier,
ResourceRetentionStatus
)

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_N2N_HO_Complete request message.
Status	STATUS	Status of operation.
MNIdentifier	MIHF_ID	This identifies the MIHF of MN.
SourceLinkIdentifier	LINK_TUPLE_ID	Specifies the source link of the handover.
TargetLinkIdentifier	LINK_TUPLE_ID	Specifies the target link of the handover.
ResourceRetention-Status	LINK_RR_STATUS	Status of local resource at the invoker of this primitive, which can be either the source network or the target network depending on the handover flow. Note, this parameter is not included if Status does not indicate "Success".

7.4.24.3.3 When generated

The MIH User responds with this primitive after processing the handover complete request.

7.4.24.3.4 Effect on receipt

Upon receipt, the local MIHF sends an MIH_N2N_HO_Complete response message to the destination MIHF.

7.4.24.4 MIH_N2N_HO_Complete.confirm

7.4.24.4.1 Function

This primitive is used by the MIHF to confirm that an MIH_N2N_HO_Complete response message is received from a peer MIHF.

7.4.24.4.2 Semantics of service primitive

```
MIH_N2N_HO_Complete.confirm (
                SourceIdentifier,
                Status,
                MNIdentifier
                SourceLinkIdentifier,
                TargetLinkIdentifier,
                ResourceRetentionStatus
                )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_N2N_HO_Complete response message.

Status	STATUS	Status of operation.
MNIdentifier	MIHF_ID	This identifies the MIHF of MN.
SourceLinkIdentifier	LINK_TUPLE_ID	Specifies the source link of the handover.
TargetLinkIdentifier	LINK_TUPLE_ID	Specifies the target link of the handover.
ResourceRetention-Status	LINK_RR_STATUS	Status of local resource at the invoker of MIH_N2N_HO_Complete.resonse primitive, which can be either the source network or the target network depending on the handover flow. Note, this parameter is not included if Status does not indicate “Success”.

7.4.24.4.3 When generated

This primitive is generated by MIHF on receiving MIH_N2N_HO_Complete response message from a peer MIHF.

7.4.24.4.4 Effect on receipt

Upon receipt, the MIH User determines that the handover complete request was processed successfully. However, if Status does not indicate “Success”, the recipient ignores any other returned values and, instead, performs appropriate error handling.

7.4.25 MIH_Get_Information

7.4.25.1 MIH_Get_Information.request

7.4.25.1.1 Function

This primitive is used by an MIH User to request information from an MIH 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 802.11 and 802.16.

7.4.25.1.2 Semantics of service primitive

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

Parameters:

Name	Type	Description
Destination Identifier	MIHF_ID	The local MIHF or a remote MIHF that will 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 C-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 C-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 MIH_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 Info Query Binary Data List and Info Query RDF Data List.
UnauthenticatedInformation-Request	BOOLEAN	The value of UIR bit to be set in the MIH_Get_Information request message sent to the remote MIHF.

One and only one of the following parameters is specified:

— Info Query Binary Data List

— Info Query RDF Data List

— Info Query RDF Schema URL

— Info Query RDF Schema List

7.4.25.1.3 When generated

This primitive is generated by an MIH User that is seeking to retrieve information.

The order of the queries in each of Info Query Binary Data List, Info Query RDF Data List and Info Query RDF Schema List parameters identifies the priority of the query. The first query has the highest priority to be processed by MIIS.

7.4.25.1.4 Effect on receipt

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

7.4.25.2 MIH_Get_Information.indication

7.4.25.2.1 Function

This primitive is used by the MIHF to indicate that an MIH_Get_Information Request message is received from a peer MIHF.

7.4.25.2.2 Semantics of service primitive

MIH_Get_Information.indication (
SourceIdentifier,
InfoQueryBinaryDataList,
InfoQueryRDFDataList,
InfoQueryRDFSchemaURL,
InfoQueryRDFSchemaList,
MaxResponseSize,
QuerierNetworkType,
UnauthenticatedInformationRequest
)

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that sent the MIH_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 C-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 MIH_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 Info Query Binary Data List and Info Query RDF Data List.
UnauthenticatedInformation-Request	BOOLEAN	The value of UIR bit contained in the MIH_Get_Information request message received from the remote MIHF.

7.4.25.2.3 When generated

This primitive is generated by the MIHF on receiving an MIH_Get_Information request message from a peer MIHF. The order of the queries in each of Info Query Binary Data List, Info Query RDF Data List and Info Query RDF Schema List 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.

7.4.25.2.4 Effect on receipt

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

7.4.25.3 MIH_Get_Information.response

7.4.25.3.1 Function

This primitive is used by an MIH User, i.e., MIIS Server, to respond to an MIH_GET_Information.indication primitive.

7.4.25.3.2 Semantics of service primitive

```
MIH_Get_Information.response (
                DestinationIdentifier,
                Status,
                InfoResponseBinaryDataList,
                InfoResponseRDFDataList,
                InfoResponseRDFSchemaURLList,
                InfoResponseRDFSchemaList
                )
```

Parameters:

Name	Type	Description
DestinationIdentifier	MIHF_ID	The local MIHF or a remote MIHF that will 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 will be sorted from most preferred first to least preferred last.
InfoResponseRDFDataList	LIST(IR_RDF_DATA)	(Optional) A list of RDF query responses. The list will be sorted from most preferred first to least preferred last.

InfoResponseRDFSchemaURLList	LIST(IR_SCHM_URL)	(Optional) A list of RDF Schema URL. The list will 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 will be sorted from most preferred first to least preferred last.

7.4.25.3.3 When generated

This primitive is generated by an MIH User in response to a received MIH_Get_Information indication message. When the size of the Info Response parameters exceeds the maximum size specified in the Max Response Size parameter from MIH_Get_Information.indication primitive, one or more of the lower order list elements in Info Response parameters must be omitted.

7.4.25.3.4 Effect on receipt

The recipient will return an MIH_Get_Information response message to the designated MIIS client.

7.4.25.4 MIH_Get_Information.confirm

7.4.25.4.1 Function

This primitive is used by the MIHF to respond to an MIH_GET_Information.request primitive.

7.4.25.4.2 Semantics of service primitive

```
MIH_Get_Information.confirm     (
                                SourceIdentifier,
                                Status,
                                InfoResponseBinaryDataList,
                                InfoResponseRDFDataList,
                                InfoResponseRDFSchemaURLList,
                                InfoResponseRDFSchemaList
                                )
```

Parameters:

Name	Type	Description
SourceIdentifier	MIHF_ID	Specifies the MIHF ID of the node that invoked MIH_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 will be sorted from most preferred first to least preferred last.
InfoResponseRDFDataList	LIST(IR_RDF_DATA)	(Optional) A list of RDF query responses. The list will be sorted from most preferred first to least preferred last.

InfoResponseRDFSchemaURLList	LIST(IR_SCHM_URL)	(Optional) A list of RDF Schema URL. The list will 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 will be sorted from most preferred first to least preferred last.

7.4.25.4.3 When generated

This primitive is generated by the MIHF on receiving an MIH_Get_Information Response message from a peer MIHF.

7.4.25.4.4 Effect on receipt

The MIH 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 Max Response Size parameter from MIH_Get_Information.request primitive, one or more of the lower order list elements in Info Response parameters must be omitted.

7.5 MIH_NET_SAP primitives

7.5.1 MIH_TP_Data

The primitives associated with data transfers are as follows:

- MIH_TP_Data.request

- MIH_TP_Data.indication

- MIH_TP_Data.confirm

The MIHF uses the MIH_TP_Data.request primitive to request that an MIH PDU be transported. The transport service provider uses the MIH_TP_Data.indication primitive to indicate the arrival of an MIH PDU. MIH_TP_Data.confirm primitive is used to acknowledge the successful transfer of the MIH PDU.

7.5.1.1 MIH_TP_Data.request

7.5.1.1.1 Function

This primitive is the request for transfer of an MIH PDU.

7.5.1.1.2 Semantics

```
MIH_TP_Data.request      (
                         TransportType,
                         SourceAddress,
                         DestinationAddress,
                         ReliableDeliveryFlag,
                         MIHProtocolPDU
```

)

Parameters:

Name	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 MIHF
DestinationAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Destination MIHF
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.
MIHProtocolPDU	OCTET_STRING	MIH Protocol PDU to be transferred

7.5.1.1.3 When generated

This primitive is used to request that an MIH PDU be transported to a remote MIHF.

7.5.1.1.4 Effect on receipt

The receipt of this primitive causes the selected transport service provider to attempt to transport the MIH PDU.

7.5.1.2 MIH_TP_Data.indication

7.5.1.2.1 Function

This primitive is the indication of a received MIH PDU.

7.5.1.2.2 Semantics

```
MIH_TP_Data.indication  (
                        TransportType,
                        SourceAddress,
                        DestinationAddress,
                        ReliableDeliveryFlag,
                        MIHProtocolPDU
                        )
```

Parameters:

Name	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 MIHF

DestinationAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Destination MIHF
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.
MIHProtocolPDU	OCTET_STRING	MIH Protocol PDU received.

7.5.1.2.3 When generated

This primitive is used by the transport service provider to indicate that an MIH PDU has been received from a remote MIHF.

7.5.1.2.4 Effect on receipt

The receipt of this primitive causes the MIHF to receive the MIH PDU that was transported.

7.5.1.3 MIH_TP_Data.confirm

7.5.1.3.1 Function

This primitive is used to confirm an acknowledged transfer.

7.5.1.3.2 Semantics

```
MIH_TP_Data.confirm     (
                        Status,
                        TransportType,
                        SourceAddress,
                        DestinationAddress
                        )
```

Parameters:

Name	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 MIHF
DestinationAddress	TRANSPORT_ADDR	Protocol layer specific Transport Address of entity that has the Destination MIHF

7.5.1.3.3 When generated

This primitive is passed from the transport service provider to the MIHF to confirm that a request to transfer an MIH PDU succeeded.

7.5.1.3.4 Effect on receipt

Upon receipt of this primitive, the receiving MIHF stops its retransmission timer for the corresponding request. When the MIHF does not receive this primitive for a pre-defined time after transmitting an MIH_TP_Data.request with ReliableDeliveryFlag set to TRUE, the MIHF attempts to retransmit the MIH_TP_Data.request.

8. Media independent handover protocol

8.1 Introduction

The MIH Function entities in MN and network entities communicate with each other using the MIH protocol messages specified in Clause 8. The MIH protocol defines message formats for exchanging these messages between peer MIH Function entities. These messages are based on the primitives that are part of the MIH Services.

8.2 MIH protocol description

8.2.1 MIH protocol transaction

The Media Independent Handover protocol defines a message exchange between two MIHF entities to support remote MIHF services. An MIH transaction is identified by a sequence of messages with the same Transaction-ID submitted to, or received from, one specific remote MIHF ID.

At any given moment, an MIH node shall have no more than one transaction pending for each direction with a certain MIH peer. In other words, the MIH node shall wait until any pending outgoing transaction is completed before it creates another outgoing transaction for the same peer. Similarly, the MIH node shall wait until any pending incoming transaction is completed before it creates another incoming transaction for the same peer.

8.2.2 MIH protocol acknowledgement service

The acknowledgement service shall be used when the MIH transport used for remote communication does not provide reliable services. When the MIH 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 MIH transport is reliable, the acknowledgement service is optional.

The source MIHF requests for an acknowledgement message to ensure successful receipt of an MIH protocol message. This MIH message is used to acknowledge the successful receipt of an MIH protocol message at the destination MIHF.

The MIH acknowledgement service is supported by the use of two bits of information that are defined exclusively for acknowledgement (ACK) usage in the MIH header. The ACK-Req bit is set by the source MIH node and the ACK-Rsp bit is set by the destination MIH node to utilize the acknowledgement service. It is expected that the underlying transport layer would take care of ensuring the integrity of the MIH protocol message during delivery.

When seeking acknowledgement service, the source MIH node shall start a retransmission timer after sending an MIH protocol message with the ACK-Req bit set and saves a copy of the MIH 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 MIH 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 MIH node receives the acknowledgement before the expiration of the timer on the first or any subsequent retransmitted attempt, then the source MIH node has ensured the receipt of the MIH packet and therefore, resets the timer and releases the saved copy of the MIH protocol message. During retransmission, if the source MIH node receives the acknowledgement for any of the previous transmission attempts then the source MIH node determines successful delivery of the message and does not have to wait for any further acknowledgements for the current message. The source MIH node retransmits an MIH 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 can be configured through a parameter defined in the MIB, see Annex E. The source MIH 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 MIH message. Implementations may consider adjusting the retransmission time-out (RTO) when operating over links with power save mobile nodes.

When a destination MIH node receives an MIH protocol message with the ACK-Req bit set, then the destination MIH node returns an MIH message with the ACK-Rsp bit set and copying the Message-ID and Transaction-ID from the received MIH protocol message. The MIH message with the ACK-Rsp bit set has only the MIH header and no other payload. In instances where the destination MIH node immediately processes the received MIH protocol message and a response is immediately available, then the ACK-Rsp bit is set in the corresponding MIH protocol response message.

The destination MIH node responds with an acknowledgement message for duplicate MIH messages (messages with same transaction-ID) that have the ACK-Req bit set. However, the destination MIH node does not process these duplicate messages if it has already done so. If a destination MIH node receives an MIH protocol message with no ACK-Req bit set then no action is taken with respect to the acknowledgement service.

In all cases, the MIH protocol message in a transaction is processed only once at the destination MIH node, irrespective of the number of received messages with the ACK-Req bit set. The destination MIH node sets the ACK-Rsp bit in an MIH protocol response message and additionally requests acknowledgement by setting the ACK-Req bit for the same MIH protocol response message.

8.2.3 MIH 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 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 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 MIHF by an instance of the transaction source or destination state machine. Optionally, each transaction can also have 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 can only be accessed 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.

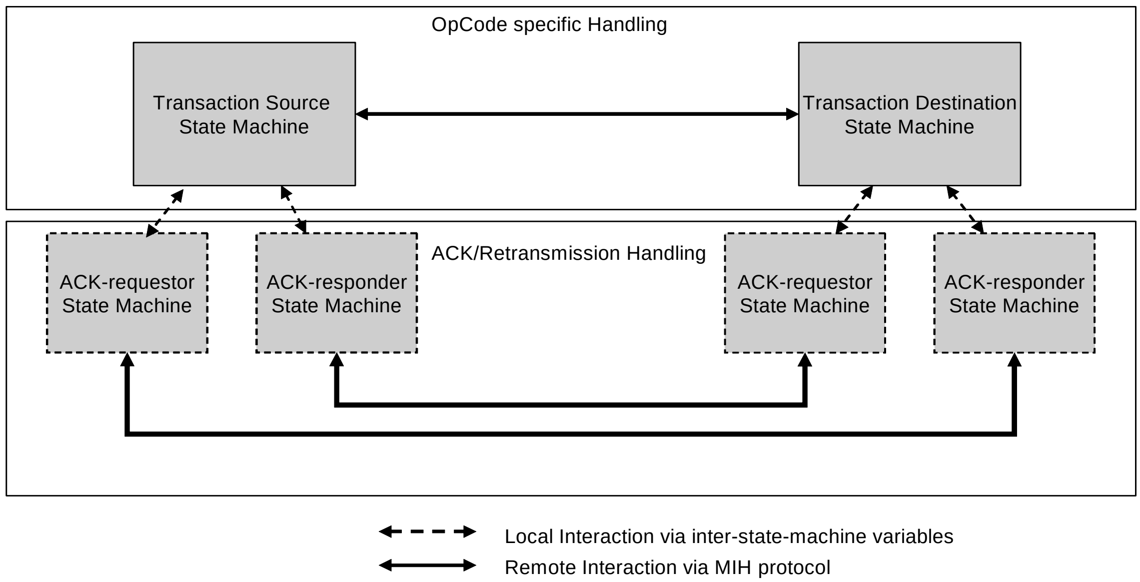


Figure 21—State machines interactions

8.2.3.2 Notational conventions used in state diagrams

State diagrams are used to represent the operation of an MIH transaction as a group of connected, mutually exclusive states. At any given time, only one state of each state machine can be 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 the 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 18; these symbols and operators are derived from the notation of the "C" programming language, ANSI X3.159.

Table 18—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.
if...then...	Conditional action. If the Boolean expression following the **if** evaluates to TRUE, then the action following the **then** is executed.
{statement 1, ... statement N}	Compound statement. Braces are used to group statements that are executed together as if they were a single statement.
!=	Inequality. Evaluates to TRUE if the expression to the left of the operator is not equal in value to the expression to the right.
==	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.
>	Greater than. Evaluates to TRUE if the value of the expression to the left of the operator is greater than the value of the expression to the right.
>=	Greater than or equal to. Evaluates to TRUE if the value of the expression to the left of the operator is either greater than or equal to the value of the expression to the right.
+	Arithmetic addition operator.
++	Arithmetic increment by one operator.
-	Arithmetic subtraction 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 19 and Table 20, respectively.

Table 19—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 20—Exported state machine variables

Name	Type	Description
TID	TID	A transaction identifier.
MyMihfID	MIHF_ID	The MIHF ID of this MIH node.
PeerMihfID	MIHF_ID	The MIHF ID of the peer MIH node.
MsgIn	MIH_MESSAGE	A valid incoming message received from a remote MIHF. 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).

Name	Type	Description
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 22. The correspondence between an incoming message and a transaction is based on TID, MyMihfID and PeerMihfID 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 MIH node has no more than one transaction pending for each direction with a certain MIH peer.
MsgOut	MIH_MESSAGE	A valid outgoing message generated by the local MIHF to be sent to the remote MIHF. 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, MyMihfID and PeerMihfID 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 MIH node has no more than one transaction pending for each direction with a certain MIH peer.

Name	Type	Description
TransactionStatus	ENUMERATED	Indicates the status of the transaction. This variable is written by the state machine and read by the MIHF. 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 21—State Machines to be searched for outgoing message

Operation code	Destination identifier TLV	State machine instances to be searched: transaction source state machine (S) or transaction destination state machine (D)
Request (0x1) / Indication (0x3)	-	S
Response (0x2)	Broadcast	S
	Unicast	D

Table 22—State Machines to be searched for incoming message

Operation code	Destination identifier TLV	ACK-Rsp bit	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)	Broadcast	-	D
	Unicast	0	S
		1	D

8.2.3.4 Inter-state-machine procedures

a) **BOOLEAN Process(MIH_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(MIH_MESSAGE)** - This procedure transmits the message passed as the input variable.

c) **BOOLEAN IsBroadcastMsg(MIH_MESSAGE)**- This procedure outputs TRUE if the input message has a broadcast destination MIHF ID. Otherwise outputs FALSE.

d) **MIHF_ID SrcMIHF_ID(MIH_MESSAGE)** - This procedure obtains a Source Identifier TLV from the message passed as the input and returns the value of the TLV.

e) **MIHF_ID DstMIHF_ID(MIH_MESSAGE)** - This procedure obtains a Destination Identifier TLV from the message passed as the input and returns the value of the TLV.

f) **void SetMIHF_ID(MIH_MESSAGE, MIHF_ID, MIHF_ID)** - This procedure inserts a Source Identifier TLV and a Destination Identifier TLV into the MIH message. The first MIHF_ID is used as the value of the Source Identifier TLV. The second MIHF_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

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.

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.

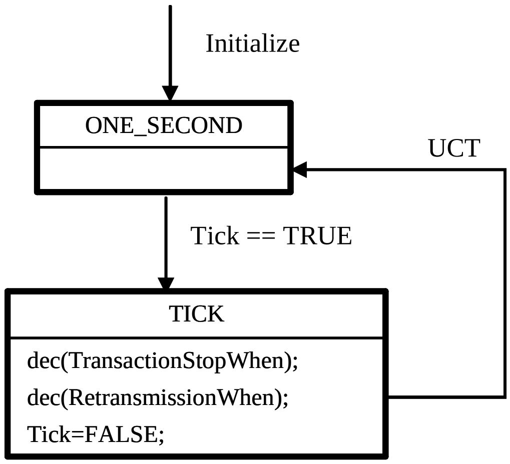


Figure 22—Transaction timers state machine

8.2.3.7 Transaction source and destination state machines

8.2.3.7.1 Intra-state-machine variables

IsBroadcast - This variable's type is BOOLEAN. When its value is TRUE it indicates that a message has a broadcast destination MIHF ID. Otherwise, its value is FALSE.

8.2.3.7.2 Intra-state-machine procedures

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.

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 can cease to exist once the value of TransactionStatus is set to either SUCCESS or FAILURE.

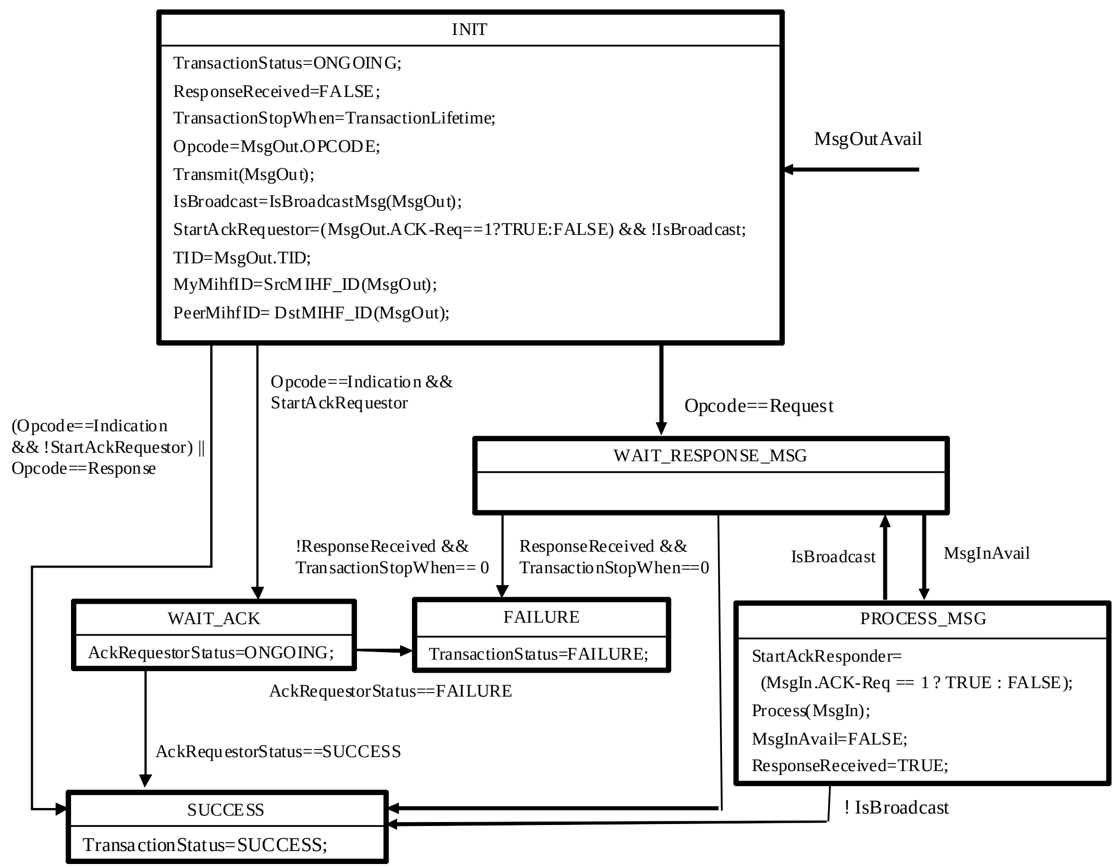


Figure 23—Transaction source state machine

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 can cease to exist once the value of TransactionStatus is set to either SUCCESS or FAILURE.

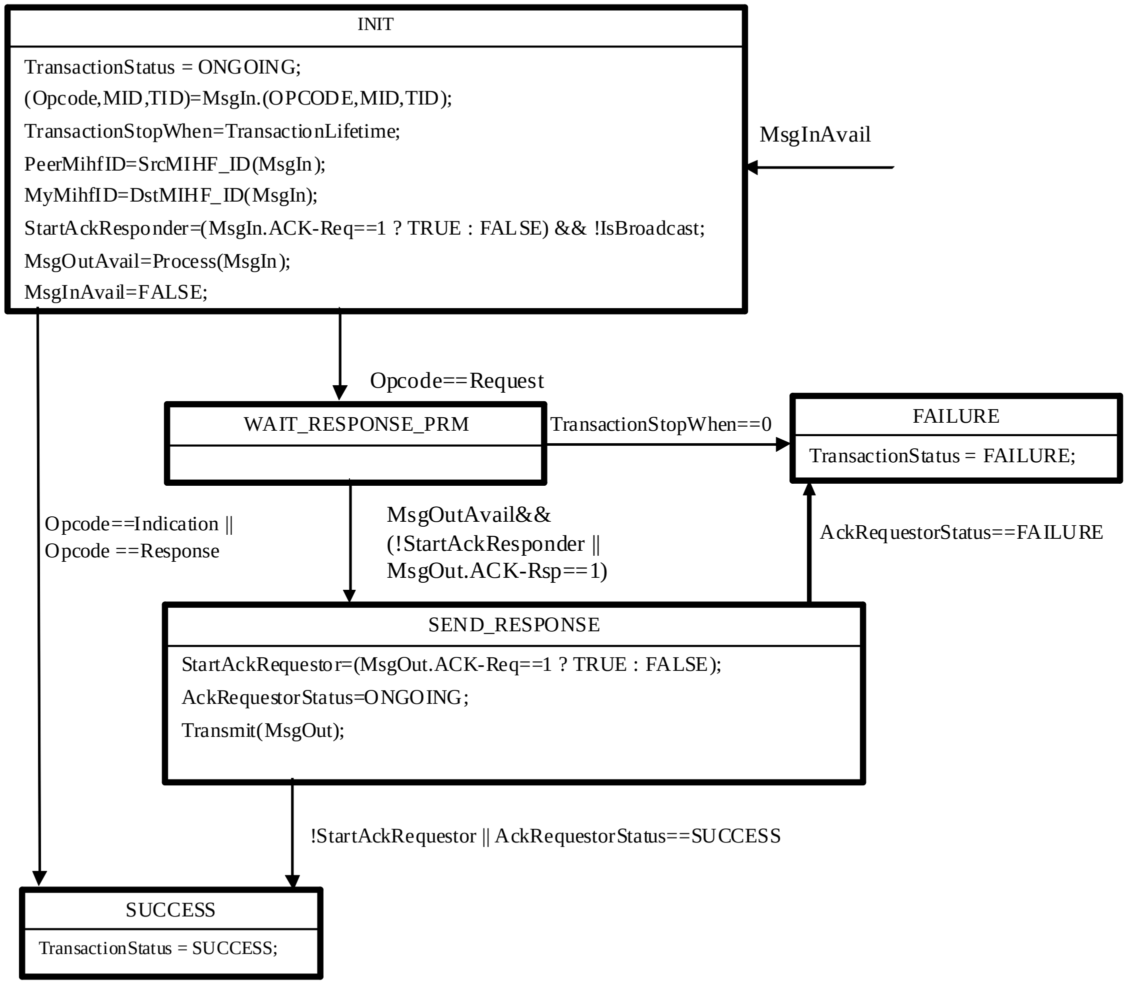


Figure 24—Transaction destination state machine

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

DUP - This variable is of type MIH_MESSAGE and represents an MIH message that has already been sent. This variable is used within ACK Responder state machine.

ACK - This variable is of type MIH_MESSAGE and represents an MIH message with the ACK-Rsp bit set and the same message ID and transaction ID as the MIH message it acknowledges. This variable is used within ACK Responder state machine.

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

RetransmissionInterval - The time interval between two subsequent transmissions of a specific message.

MaxRtxCtr - The maximum number of times that a message will be retransmitted, if retransmission conditions occur.

The maximum number of retransmissions and the retransmission interval depends on the characteristics of the underlying transport. These configuration parameters are defined in a MIB, see Annex F.

Note that the maximum number of retransmission 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 can cease 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.

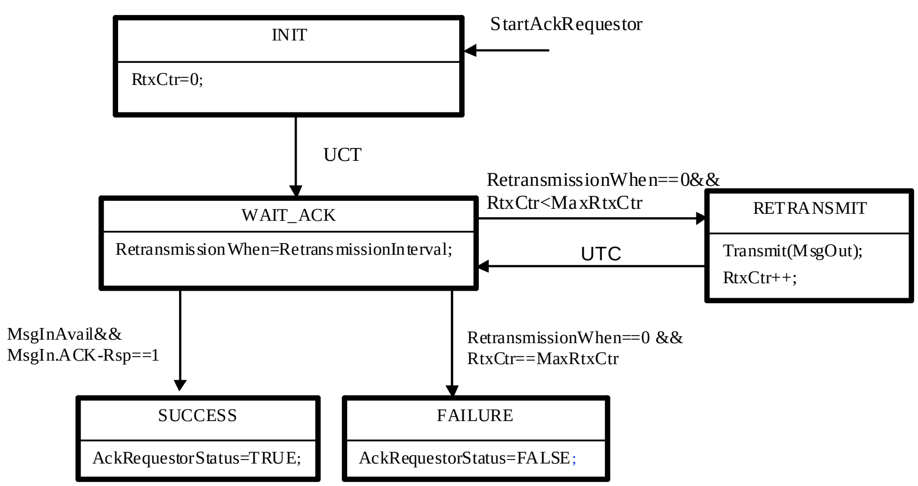


Figure 25—ACK requestor state machine

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 can cease to exist once its associated transaction source or transaction destination state machine ceases to exist.

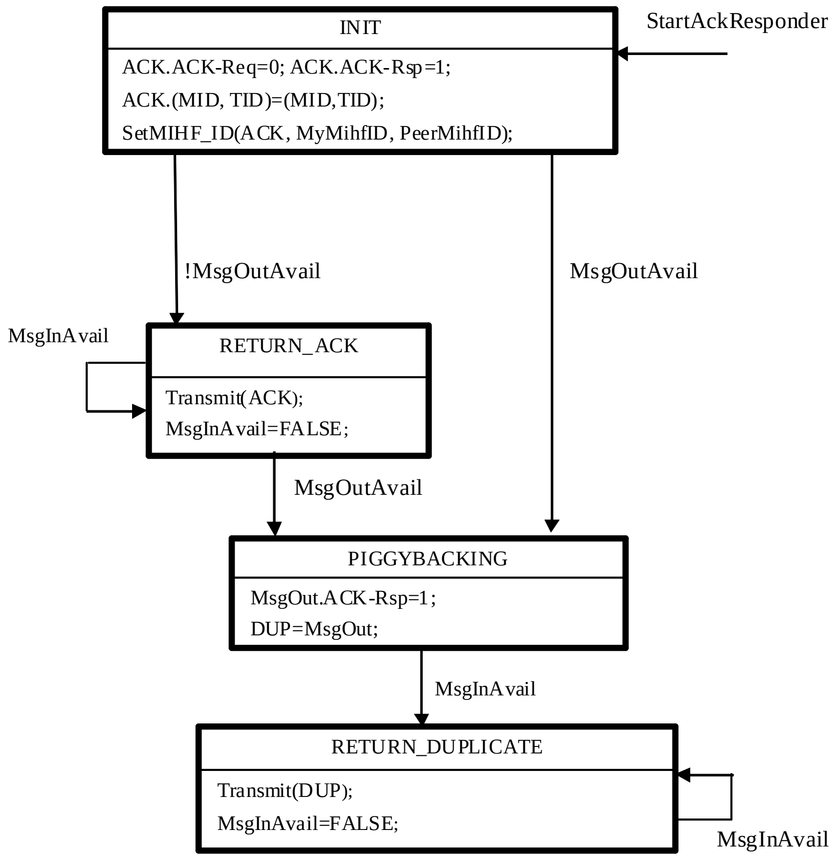


Figure 26—ACK responder state machine

8.2.4 Other considerations

8.2.4.1 Congestion control and load management

The MIH protocol does not provide direct support for congestion control. Therefore, it is recommended to run the MIH protocol over congestion aware transport layers.

In order to help prevent congestion, flow control mechanisms are implemented at the MIHF. 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, MIH 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 MIHF suffers from overload, it drops requests from MIH 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 MIHF invokes flow control towards a specific requestor when overloaded with reliably delivered messages.

8.2.4.2 Reliability

MIH protocol messages are delivered via media dependent transport. To 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 may be implemented.

A reliable media dependent transport is one which exhibits a message loss rate of less than 0.01%,

8.2.4.3 MIHF discovery

8.2.4.3.1 General

The MIHF discovery refers to the procedure that allows one MIHF to discover its peer MIHFs (e.g., an MN discovers available peer MIHFs in an access network). MIHF discovery can be done either at layer 2 or layer 3. However, MIHF discovery at layer 3 and MIHF discovery performed over the control plane using media specific broadcast control message are outside the scope of this standard.

8.2.4.3.2 Combined MIH function discovery and capability discovery over data plane

Combined MIH function discovery and capability discovery is performed to discover the MIHF ID, the peer MIHF transport address, and MIHF capabilities at the same time. As stated in 6.2.3, MIHF Discovery can be implicitly performed using the MIH Capability Discovery when both MIH nodes are residing in the same broadcast domain. If MIHF ID and transport address are known, e.g., pre-configured, MIHF uses MIH_Capability_Discover messages to discover MIHF capabilities only. The following subclauses refer to the MIH Capability Discovery both as a means to discover the MIHF and its capabilities.

8.2.4.3.3 Unsolicited MIH capability discovery

An MIHF discovers peer MIHF entities and their capabilities either by listening to media-specific broadcast messages or media independent MIH capability broadcast messages.

For example, by listening to a media-specific broadcast message such as a Beacon frame in IEEE 802.11 or a DCD in IEEE 802.16, link layers on an MN forward the received message to its MIHF. An MIHF receives an MIH_Capability_Discover response message broadcasted over the data plane periodically from an MIH entity.

8.2.4.3.4 Solicited MIH capability discovery

An MIHF (the requestor) discovers its peer MIH functions and its capability by broadcasting or unicasting an MIH_Capability_Discover request message to either its broadcast domain or a known MIHF ID and address, respectively. Only MIH network entities respond to a broadcasted MIH_Capability_Discover request.

When a peer MIH function (the responder) receives the MIH_Capability_Discover request message, it sends MIH_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 MIH_Capability_Discover response message, it learns the responder's MIHF ID by checking the source ID of MIH_Capability_Discover response.

For complete operation, the requestor sets a timer at the time of sending an MIH_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 MIH function and capability discovery procedure. When the timer expires without receiving a response message, the requestor tries the combined MIH function discovery and capability discovery procedure by using a different transport or terminate the MIH function and capability discovery procedure.

8.3 MIH protocol identifiers

Following identifiers are used in MIH protocol messages:

— MIHF ID

— Transaction ID

8.3.1 MIHF ID

MIHF Identifier (MIHF ID) is an identifier that is required to uniquely identify an MIHF entity for delivering the MIH services. MIHF ID is used in all MIH protocol messages. This enables the MIH protocol to be transport agnostic.

MIHF ID is assigned to the MIHF during its configuration process. The configuration process is outside the scope of the standard.

Broadcast MIHF ID is defined as an MIHF ID of zero length. A broadcast MIHF ID can be used when destination MIHF ID is not known to a sending MIHF. When MIH protocol message with broadcast MIHF ID is transmitted over data plane, the MIH protocol message is broadcasted over either L2 or L3 data plane.

The MIHF ID is of type MIHF_ID (See Annex C.3.11).

8.3.2 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 MIH protocol frame.

Transaction ID is defined as a 16 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 2^16) every time a new Transaction ID is generated.

8.4 MIH protocol frame format

8.4.1 General frame format

In MIH protocol messages, all TLV definitions are always aligned on an octet boundary and hence no padding is required. An MIH protocol payload carries a Source MIHF Identifier TLV and a Destination MIHF Identifier TLV followed by MIH Service Specific TLVs.

Figure 27 shows the components of the MIH protocol frame.

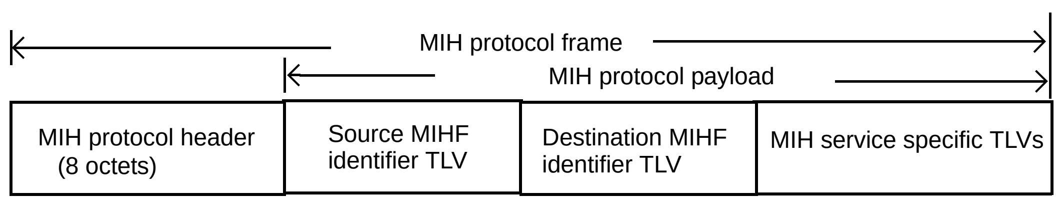


Figure 27 — MIH protocol general frame format

8.4.1.1 MIH protocol header fields

The MIH protocol header (see Figure 28) carries the essential information that is present in every frame and is used for parsing and analyzing the MIH protocol frame.

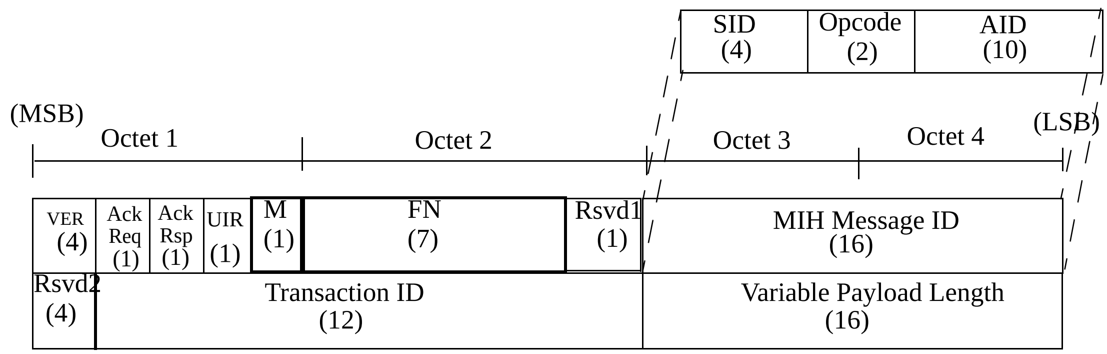


Figure 28 — MIH protocol header format

Table 23 shows the description of the header fields.

Table 23—Description of MIH protocol header fields

Field name	Size (bits)	Description
Version	4	This field is used to specify the version of MIH protocol used. 0: Not to be used 1: First version 2 - 15: (*Reserved*) The version number will be incremented only when a fundamental incompatibility exists between a new revision and the prior edition of the standard. An MIH node that receives an MIH message with a higher version number than it supports will discard the frame without indication to the sending MIH 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 MIH Information Service to indicate if the protocol message is sent in pre-authentication/pre-association state so that the length of the response message can be limited. The UIR bit should be set to '1' by the originator when making an MIH information service request over a certain link in the un-associated/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.

Table 23—Description of MIH protocol header fields

Field name	Size (bits)	Description
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, all the bits of this field are to be set to '0'.
MIH message ID (MID)	16	Combination of the following 3 fields.
-- Service identifier (SID)	4	Identifies the different MIH services, possible values are: 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: 1: Request 2: Response 3: Indication
-- Action identifier (AID)	10	This indicates the action to be taken with regard to the SID (see Table D-1 for AID assignments).
Reserved2	4	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 MIH protocol frame. The length of the MIH protocol header is NOT included.

8.4.2 Fragmentation and reassembly

8.4.2.1 General

The MIH fragmentation mechanism is defined using 'M' (More Fragment) and 'FN' (Fragment Number) fields of the MIH protocol header.

An MIH message is fragmented only when MIH message is sent natively over a 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 aFragmentationThreshold, which shall be equal to the Maximum Transmission Unit (MTU) of the link layer that is on the path between two MIHF nodes. When the MTU of the link layer between two MIHF nodes is known, the maximum fragment size is set to the MTU. The method of determining such an MTU is outside the scope of this standard. When the MTU of the link layer between two MIHF nodes is unknown, the maximum fragment size is set to the minimum MTU of 1500 octets. When MIH message is sent using a L3 or higher layer transport, L3 takes care of any fragmentation issue and the MIH protocol does not handle fragmentation in such cases.

Figure 29 shows the components of the fragmented MIH protocol frame. The MIH protocol payload carries a Source MIHF Identifier TLV and a Destination MIHF Identifier TLV followed by a fragment payload.

Based on the fragment size, the fragment payload may not be aligned on a TLV boundary, i.e., TLVs other than the source MIHF identifier and destination MIHF identifier TLVs may not be complete within the fragment payload. The fragment size may be smaller than the maximum fragment size and shall be larger than that can generate more than 128 fragments.

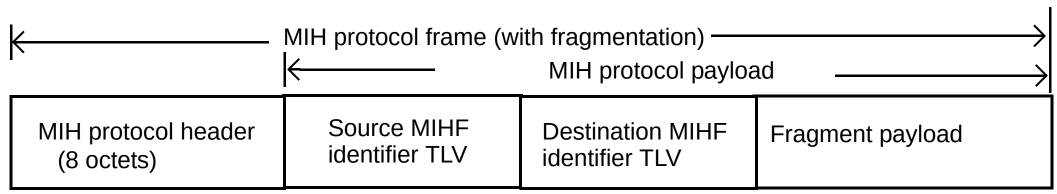


Figure 29—Fragmented MIH protocol frame format

8.4.2.2 Fragmentation

When an MIH message is fragmented, the fragmentation is performed within 'Transmit()' procedure in the MIH transaction protocol state machines. The MIH protocol header, the source MIHF identifier TLV and destination MIHF identifier TLV of the original message are copied to each fragment. 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 MIH protocol payload of that fragment.

'More fragment' and 'fragment number' fields of each fragment are set according to the description in Table 23.

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 MIHF.

No retransmission by the MIH protocol (defined in 8.2) is performed for any single fragment of a multi-fragment message.

8.4.2.3 Reassembly

The destination MIHF reassembles the received fragments into an original message. Reassembly is performed outside the MIH transaction state machines. 'MsgIn' and 'MsgInAvail' variables are set only after successful reassembly. An MIHF shall be capable of receiving fragments of arbitrary length.

The following fields are used for reassembling fragments:

- MIH message ID
- Transaction ID
- Source MIHF identifier TLV
- Destination MIHF identifier TLV
- More fragment
- Fragment number

When any fragment of a multi-fragment message has arrived first, the destination MIHF starts a timer referred to as ReassemblyTimer. If this ReassemblyTimer expires before all fragments have been received, the destination MIHF discards those fragments that it has received. A duplicate fragment is discarded.

An example of an original MIH message and fragmented MIH messages is shown in Annex M.

8.5 Message parameter TLV encoding

The following TLV encoding shall be used for all parameters in an MIH 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.

Specifically, the *Type* field is one octet[2], and the *Length* shall be encoded with the rules described in 6.5.6.2.

Moreover, TLV *Type* values shall be unique within the MIH protocol. The TLV encoding starts at 1 and any subsequent values are assigned in ascending order.

8.6 MIH protocol messages

The following subclauses specify different MIH protocol messages in TLV form. The shaded areas represent the MIH protocol header, while the unshaded areas represent the MIH protocol payload. The payload consists of a set of identifiers in TLV form.

The TLV Type assignment for each TLV can be found in Annex C.

TLV type values ranging from 110 to 127 are reserved for experimental TLVs. These values are used by different implementations to evaluate the option of using TLVs not defined by the specification.

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 MIHF information that is communicating with another MIHF with different experimental TLVs implementation.

All MIH messages carry a source MIHF ID followed by a destination MIHF ID as the first two TLVs of the MIH protocol payload part of the message. Broadcast MIHF ID is allowed in an MIH_Capability_Discover request and response message as its destination MIHF ID.

All “Optional” fields are optionally sent but the receiver shall properly operate on them if present, i.e. these fields are not optional in the implementation, but only in their use.

On receipt of an MIH request message the MIHF shall respond with a corresponding response message.

Any message received that has an invalid MIH header, or does not contain the source/destination MIHF IDs, or has an unrecognizable or invalid MIH Message ID shall be discarded without sending any indication to the source MIH node. Any undefined or unrecognizable TLVs in a received message shall be ignored by the receiver.

[2]Note that the TLV *Type* field length is different than the Information Element *Type* length, which is four octets.

8.6.1 MIH messages for service management

8.6.1.1 MIH_Capability_Discover request

The corresponding MIH primitive of this message is defined in 7.4.1.1.

If a requesting MIHF entity does not know the destination MIHF entity's MIHF ID, the requesting MIHF entity fills its destination MIHF ID with broadcast MIHF ID and broadcasts this message over the data plane, either L2 or L3. If a requesting MIHF entity knows the destination MIHF entity's MIHF ID, the requesting MIHF entity fills its destination MIHF ID and unicast this message over the data plane, either L2 or L3.

If the generation of this message is invoked upon receiving MIH capability advertisement in unauthenticated state through media specific broadcast message, such as Beacon frame and DCD, destination MIHF ID is filled with broadcast MIHF ID and this message is transmitted over the control plane using a L2 management frame, such as a 802.11 management action frame or a 802.16 MIH MAC management message.

This message contains the SupportedMihEventList, SupportedMihCommandList, SupportedISQueryTypeList, SupportedTransportList, and MBBHandoverSupport TLVs to enable the receiving MIHF to discover the sending MIHF's capability. Therefore, peer MIHF entities can discover each other's MIH capability by one MIH protocol message transaction.

MIH Header Fields (SID=1, Opcode=1, AID=1)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkAddressList (optional) (Link address list TLV)
SupportedMihEventList (optional) (MIH event list TLV)
SupportedMihCommandList (optional) (MIH command list TLV)
SupportedISQueryTypeList (optional) (MIIS query type list TLV)
SupportedTransportList (optional) (Transport option list TLV)
MBBHandoverSupport (optional) (MBB handover support TLV)

8.6.1.2 MIH_Capability_Discover response

The corresponding MIH primitive of this message is defined in 7.4.1.3.

Only an MIHF capable network entity responds with an MIH_Capability_Discover response to the received MIH_Capability_Discover request with a broadcast MIHF ID, or send unsolicited MIH_Capability_Discover responses periodically. When an MIH network entity broadcasts an unsolicited

MIH_Capability_Discover response to advertise its MIHF ID and capabilities, Destination ID is a broadcast MIHF ID.

MIH Header Fields (SID=1, Opcode=2, AID=1)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
Link Address List (optional) (Link address list TLV)
SupportedMihEventList (optional) (MIH event list TLV)
SupportedMihCommandList (optional) (MIH command list TLV)
SupportedISQueryTypeList (optional) (MIIS query type list TLV)
SupportedTransportList (optional) (Transport option list TLV)
MBBHandoverSupport (optional) (MBB handover support TLV)

8.6.1.3 MIH_Register request

The corresponding MIH primitive of this message is defined in 7.4.2.1.

This message is transmitted to the remote MIHF to perform a registration or re-registration.

MIH Header Fields (SID=1, Opcode=1, AID=2)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkIdentifierList (Link identifier list TLV)
RequestCode (Register request code TLV)

8.6.1.4 MIH_Register response

The corresponding MIH primitive of this message is defined in 7.4.2.3.

This message is sent in response to a registration or re-registration request.

MIH Header Fields (SID=1, Opcode=2, AID=2)

Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
ValidTimeInterval (not included if Status does not indicate “Success”) (Valid time interval TLV)

8.6.1.5 MIH_DeRegister request

The corresponding MIH primitive of this message is defined in 7.4.3.1.

This message is transmitted to the remote MIHF to request a de-registration. There is no parameter for this message.

MIH Header Fields (SID=1, Opcode=1, AID=3)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)

8.6.1.6 MIH_DeRegister response

The corresponding MIH primitive of this message is defined in 7.4.3.3.

This message is sent in response to a de-registration request.

MIH Header Fields (SID=1, Opcode=2, AID=3)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)

8.6.1.7 MIH_Event_Subscribe request

The corresponding MIH primitive of this message is defined in 7.4.4.1.

This message is sent by a remote MIHF (the subscriber) to subscribe to one or more event types from a particular event origination point.

MIH Header Fields (SID=1, Opcode=1, AID=4)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)

Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkIdentifier (Link identifier TLV)
RequestedMihEventList (MIH event list TLV)
EventConfigurationInfoList (optional) (Event configuration info list TLV)

8.6.1.8 MIH_Event_Subscribe response

The corresponding MIH primitive of this message is defined in 7.4.4.2.

The response indicates which of the event types were successfully subscribed.

MIH Header Fields (SID=1, Opcode=2, AID=4)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
LinkIdentifier (Link identifier TLV)
ResponseMihEventList (not included if Status does not indicate “Success”) (MIH event list TLV)

8.6.1.9 MIH_Event_Unsubscribe request

The corresponding MIH primitive of this message is defined in 7.4.5.1.

This message is sent by a remote MIHF (the subscriber) to unsubscribe from a set of link layer events.

MIH Header Fields (SID=1, Opcode=1, AID=5)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkIdentifier (Link identifier TLV)
RequestedMihEventList (MIH event list TLV)

8.6.1.10 MIH_Event_Unsubscribe response

The corresponding MIH primitive of this message is defined in 7.4.5.2.

The response indicates which of the event types were successfully unsubscribed.

MIH Header Fields (SID=1, Opcode=2, AID=5)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
LinkIdentifier (Link identifier TLV)
ResponseMihEventList (not included if Status does not indicate "Success") (MIH event list TLV)

8.6.2 MIH messages for event service

8.6.2.1 MIH_Link_Detected indication

The corresponding MIH primitive of this message is defined in 7.4.6.1.

This message is transmitted to the remote MIHF when a new link has been detected.

MIH Header Fields (SID=2, Opcode=3, AID=1)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkDetectedInfoList (Link detected info list TLV)

8.6.2.2 MIH_Link_Up indication

The corresponding MIH primitive of this message is defined in 7.4.7.1.

This notification is delivered from an MIHF, when present in the PoA, to an MIHF in the network when a layer 2 connection is successfully established with an MN.

MIH Header Fields (SID=2, Opcode=3, AID=2)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkIdentifier (Link identifier TLV)
OldAccessRouter (optional) (Old access router TLV)

NewAccessRouter (optional) (New access router TLV)
IPRenewalFlag (optional) (IP renewal flag TLV)
MobilityManagementSupport (optional) (Mobility management support TLV)

8.6.2.3 MIH_Link_Down indication

The corresponding MIH primitive of this message is defined in 7.4.8.1.

This notification is delivered from an MIHF, when present in the PoA, to an MIHF in the network when a layer 2 connection with an MN is disconnected due to a certain reason.

MIH Header Fields (SID=2, Opcode=3, AID=3)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkIdentifier (Link identifier TLV)
OldAccessRouter (optional) (Old access router TLV)
ReasonCode (Link down reason code TLV)

8.6.2.4 MIH_Link_Parameters_Report indication

The corresponding MIH primitive of this message is defined in 7.4.9.1.

This message indicates changes in link conditions that have crossed pre-configured threshold levels. A pre-configured threshold level is set by the MIH_Link_Configure_Thresholds request message.

MIH Header Fields (SID=2, Opcode=3, AID=5)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkIdentifier (Link identifier TLV)
LinkParameterReportList (Link parameter report list TLV)

8.6.2.5 MIH_Link_Going_Down indication

The corresponding MIH primitive of this message is defined in 7.4.10.1.

This message is transmitted to the remote MIHF when a layer 2 connectivity is expected (predicted) to go down within a certain time interval.

MIH Header Fields (SID=2, Opcode=3, AID=6)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkIdentifier (Link identifier TLV)
TimeInterval (Time interval TLV)
LinkGoingDownReason (Link going down reason TLV)

8.6.2.6 MIH_Link_Handover_Imminent indication

The corresponding MIH primitive of this message is defined in 7.4.11.1.

This message indicates that a link layer handover decision has been made and its execution is imminent.

MIH Header Fields (SID=2, Opcode=3, AID=7)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
OldLinkIdentifier (Link identifier TLV)
NewLinkIdentifier (New link identifier TLV)
OldAccessRouter (optional) (Old access router TLV)
NewAccessRouter (optional) (New access router TLV)

8.6.2.7 MIH_Link_Handover_Complete indication

The corresponding MIH primitive of this message is defined in 7.4.12.1.

This message indicates that a link layer handover has been completed.

MIH Header Fields (SID=2, Opcode=3, AID=8)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)

OldLinkIdentifier (Link identifier TLV)
NewLinkIdentifier (New link identifier TLV)
OldAccessRouter (optional) (Old access router TLV)
NewAccessRouter (optional) (New access router TLV)
LinkHandoverStatus (Status TLV)

8.6.3 MIH messages for command service

8.6.3.1 MIH_Link_Get_Parameters request

The corresponding MIH primitive of this message is defined in 7.4.14.2.

This message is used to discover the status of currently available links.

MIH Header Fields (SID=3, Opcode=1, AID=1)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
DeviceStatesRequest (optional) (Device states request TLV)
LinkIdentifierList (Link identifier list TLV)
GetStatusRequestSet (Get status request set TLV)

8.6.3.2 MIH_Link_Get_Parameters response

The corresponding MIH primitive of this message is defined in 7.4.14.3.

This message is used by an MIHF to report the status of currently available links.

MIH Header Fields (SID=3, Opcode=2, AID=1)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF 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 MIH_Link_Configure_Thresholds request

The corresponding MIH primitive of this message is defined in 7.4.15.2.

This message is used to configure thresholds of the lower layer link.

MIH Header Fields (SID=3, Opcode=1, AID=2)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkIdentifier (Link identifier TLV)
ConfigureRequestList (Configure request list TLV)

8.6.3.4 MIH_Link_Configure_Thresholds response

The corresponding MIH primitive of this message is defined in 7.4.15.3.

This message returns the status of a thresholds configuration request. The MIHF generating this message generates MIH_Link_Parameters_Report indication message when the configured threshold is crossed.

MIH Header Fields (SID=3, Opcode=2, AID=2)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
LinkIdentifier (Link identifier TLV)
ConfigureResponseList (not included if Status does not indicate “Success”) (Configure response list TLV)

8.6.3.5 MIH_Link_Actions request

The corresponding MIH primitive of this message is defined in 7.4.16.1.

This message is used to control the behavior of a set of lower layer links.

MIH Header Fields (SID=3, Opcode=1, AID=3)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)

Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkActionsList (Link actions list TLV)

8.6.3.6 MIH_Link_Actions response

The corresponding MIH primitive of this message is defined in 7.4.16.2.

This message returns the result of an MIH_Link_Actions request.

MIH Header Fields (SID=3, Opcode=2, AID=3)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
LinkActionsResultList (not included if Status does not indicate “Success”) (Link actions result list TLV)

8.6.3.7 MIH_Net_HO_Candidate_Query request

The corresponding MIH primitive of this message is defined in 7.4.17.2.

This message is used for communication between the MIHF on an MN and the MIHF on a network. The function is used to communicate an intent of network initiated handover.

MIH Header Fields (SID=3, Opcode=1, AID=4)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
SuggestedNewLinkList (List of link PoA list TLV)
QueryResourceReportFlag (Query resource report flag TLV)

8.6.3.8 MIH_Net_HO_Candidate_Query response

The corresponding MIH primitive of this message is defined in 7.4.17.4.

This message is used for communication between the MIHF on an MN and the MIHF on a network. The function is used to respond to an intent of network initiated handover.

MIH Header Fields (SID=3, Opcode=2, AID=4)

Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
SourceLinkIdentifier (Link identifier TLV)
HandoverStatus (not included if Status does not indicate “Success”) (Handover status TLV)
PreferredLinkList (not included if Status does not indicate “Success”) (Preferred link list TLV)

8.6.3.9 MIH_MN_HO_Candidate_Query request

The corresponding MIH primitive of this message is defined in 7.4.18.1.

This message is used by an MIHF on the MN to communicate to a network MIHF, an intent to initiate a handover.

MIH Header Fields (SID=3, Opcode=1, AID=5)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
SourceLinkIdentifier (Link identifier TLV)
CandidateLinkList (List of link PoA list TLV)
QoSResourceRequirements (Handover resource query list TLV)
IPConfigurationMethods (optional) (IP address configuration methods TLV)
DHCPServerAddress (optional) (DHCP server address TLV)
FAAddress (optional) (FA address TLV)
AccessRouterAddress (optional) (Access router address TLV)

8.6.3.10 MIH_MN_HO_Candidate_Query response

The corresponding MIH primitive of this message is defined in 7.4.18.3.

This message is used by an MIHF in the network to respond to an MIH_MN_HO_Candidate_Query request message from a remote MIHF on the MN.

MIH Header Fields (SID=3, Opcode=2, AID=5)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
SourceLinkIdentifier (Link identifier TLV)
PreferredCandidateLinkList (not included if Status does not indicate "Success") (Preferred link list TLV)

8.6.3.11 MIH_N2N_HO_Query_Resources request

The corresponding MIH primitive of this message is defined in 7.4.19.1.

This message is used by an MIHF on the serving network to communicate to an MIHF on the candidate network, an intent to initiate a handover. This message is also used to retrieve IP address related information from the candidate network.

MIH Header Fields (SID=3, Opcode=1, AID=6)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
QoSResourceRequirements (Handover resource query list TLV)
IPConfigurationMethods (optional) (IP address configuration methods TLV)
DHCPServerAddress (optional) (DHCP server address TLV)
FAAddress (optional) (FA address TLV)
AccessRouterAddress (optional) (Access router address TLV)
CandidateLinkList (optional) (Link identifier list TLV)

8.6.3.12 MIH_N2N_HO_Query_Resources response

The corresponding MIH primitive of this message is defined in 7.4.19.3.

This message is used by an MIHF in the candidate network to respond to an MIH_N2N_HO_Query_Resources request message from an MIHF on the serving network. This is used to return the result of resource preparation of the impending handover and to notify the MIHF on the serving network of the link resource status and IP address related information of the candidate network.

MIH Header Fields (SID=3, Opcode=2, AID=6)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
ResourceStatus (not included if Status does not indicate "Success") (Resource status TLV)
CandidateLinkList (optional) (not included if Status does not indicate "Success") (Preferred link list TLV)
IPAddressInformationStatus (optional) (not included if Status does not indicate "Success") (IP address information status TLV)

8.6.3.13 MIH_MN_HO_Commit request

The corresponding MIH primitive of this message is defined in 7.4.20.1.

This message is used by the MIHF on the mobile node to notify the Serving PoS of the decided target network information.

MIH Header Fields (SID=3, Opcode=1, AID=7)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkType (Link type TLV)
TargetNetworkInfo (Target network info TLV)

8.6.3.14 MIH_MN_HO_Commit response

The corresponding MIH primitive of this message is defined in 7.4.20.3.

This message is used by the MIHF on the Serving PoS to respond to an MIH_MN_HO_Commit request message.

MIH Header Fields (SID=3, Opcode=2, AID=7)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)

Status (Status TLV)
LinkType (Link type TLV)
TargetNetworkInfo (Target network info TLV)

8.6.3.15 MIH_Net_HO_Commit request

The corresponding MIH primitive of this message is defined in 7.4.21.1.

This message is used by the MIHF to communicate the intent to commit to a handover request to a specific link and PoA.

MIH Header Fields (SID=3, Opcode=1, AID=8)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
LinkType (Link type TLV)
TargetNetworkInfoList (List of target network info TLV)
AssignedResourceSet (Assigned resource set TLV)
Link Action Execution Delay (Time interval TLV)
LinkActionsList (Optional) (Link actions list TLV)

8.6.3.16 MIH_Net_HO_Commit response

The corresponding MIH primitive of this message is defined in 7.4.21.3.

This message is used by the MIHF to respond to a request to commit to a handover request to a specific link and PoA.

MIH Header Fields (SID=3, Opcode=2, AID=8)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)

LinkType (not included if Status does not indicate "Success") (Link type TLV)
TargetNetworkInfo (not included if Status does not indicate "Success") (Target network info TLV)
LinkActionsResultList (Optional) (not included if Status does not indicate "Success") (Link actions result list TLV)

8.6.3.17 MIH_N2N_HO_Commit request

The corresponding MIH primitive of this message is defined in 7.4.22.1.

This message is used by the MIHF on the serving network to communicate with its peer MIHF on the selected target network. This is used to request the target network to allocate resources to an MN that is about to move toward that network link and PoA.

MIH Header Fields (SID=3, Opcode=1, AID=9)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
MNIdentifier (Mobile node MIHF ID TLV)
TargetMobileNodeLinkIdentifier (MN link ID TLV)
TargetPoA (PoA TLV)
RequestedResourceSet (Requested resource set TLV)

8.6.3.18 MIH_N2N_HO_Commit response

The corresponding MIH primitive of this message is defined in 7.4.22.3.

This message is used by the MIHF on the selected target network to communicate with its peer MIHF on the serving network. This is used to respond to the MIH_N2N_HO_Commit request message.

MIH Header Fields (SID=3, Opcode=2, AID=9)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
MNIdentifier (Mobile node MIHF ID TLV)
TargetLinkIdentifier (not included if Status does not indicate "Success") (Link identifier TLV)

AssignedResourceSet (not included if Status does not indicate "Success") (Assigned resource set TLV)

8.6.3.19 MIH_MN_HO_Complete request

The corresponding MIH primitive of this message is defined in 7.4.23.1.

This message is used by the MIHF on the MN to communicate the status of handover operation to the MIHF on the target network.

MIH Header Fields (SID=3, Opcode=1, AID=10)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
SourceLinkIdentifier (Link identifier TLV)
TargetLinkIdentifier (New link identifier TLV)
HandoverResult (Handover result TLV)

8.6.3.20 MIH_MN_HO_Complete response

The corresponding MIH primitive of this message is defined in 7.4.23.3.

This message is used by the MIHF on the target network to communicate the response following the completion of handover operation to the MN.

MIH Header Fields (SID=3, Opcode=2, AID=10)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
SourceLinkIdentifier (Link identifier TLV)
TargetLinkIdentifier (New link identifier TLV)

8.6.3.21 MIH_N2N_HO_Complete request

The corresponding MIH primitive of this message is defined in 7.4.24.1.

This message is used by the MIHF to communicate the status of handover operation.

MIH Header Fields (SID=3, Opcode=1, AID=11)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
MNIdentifier (Mobile node MIHF ID TLV)
SourceLinkIdentifier (Link identifier TLV)
TargetLinkIdentifier (New link identifier TLV)
HandoverResult (Handover result TLV)

8.6.3.22 MIH_N2N_HO_Complete response

The corresponding MIH primitive of this message is defined in 7.4.24.3.

This message is used by the MIHF to communicate the response following the completion of the handover operation. The message is used to communicate the preferred action to be taken w.r.t. resources associated with the previous connection. If the handover is successful, the resources are released.

MIH Header Fields (SID=3, Opcode=2, AID=11)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF ID TLV)
Status (Status TLV)
MNIdentifier (Mobile node MIHF ID TLV)
SourceLinkIdentifier (Link identifier TLV)
TargetLinkIdentifier (New link identifier TLV)
ResourceRetention Status (not included if Status does not indicate “Success”) (Resource retention status TLV)

8.6.4 MIH messages for information service

MIH Information service uses only two messages - MIH_Get_Information request and MIH_Get_Information response. 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 two messages are substantially different from other MIH message parameters, and are therefore separately defined in the following sub-clauses.

8.6.4.1 MIH_Get_Information request

The corresponding MIH primitive of this message is defined in 7.4.25.1.

This message is used by an MIHF to retrieve a set of Information Elements provided by the information service. A single MIH_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.

MIH Header Fields (SID=4, Opcode=1, AID=1)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF 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 MIH_Get_Information response

The corresponding MIH primitive of this message is defined in 7.4.25.3.

This is used as a response to the MIH_Get_Information request message. The total response message size shall not exceed the value indicated in the Max Response Size TLV of corresponding MIH_Get_Information request message. The order of the query response shall be in the same order as the query requests.

MIH Header Fields (SID=4, Opcode=2, AID=1)
Source Identifier = sending MIHF ID (Source MIHF ID TLV)
Destination Identifier = receiving MIHF ID (Destination MIHF 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)

Annexes

Annex A Bibliography

(informative)

[B1] 3GPP TR 43.901, Feasibility Study on Generic Access to A/Gb Interface.

[B2] 3GPP TS 23.002, Network Architecture.

[B3] 3GPP TS 23.060, General Packet Radio Service (GPRS); Service Description; Stage 2.

[B4] 3GPP TS 23.234, 3GPP system to Wireless Local Area Network (WLAN) inter-working; System description.

[B5] 3GPP TS 24.007, Mobile Radio Interface Signaling, Layer 3, General Aspects.

[B6] 3GPP TS 25.331, Radio Resource Control Specification.

[B7] 3GPP2 C.S0001-D (2004-02), Introduction to cdma2000 Standards for Spread Spectrum Systems.

[B8] 3GPP2 C.S0002-D (2004-02), Physical Layer Standard for cdma2000 Spread Spectrum Systems.

[B9] 3GPP2 C.S0003-D (2004-02), Medium Access Control (MAC) Standard for cdma2000 Spread Spectrum Systems.

[B10] 3GPP2 C.S0005-D (2004-02), Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems.

[B11] 3GPP2 C.S0006-D (2004-02), Analog Signaling Standard for cdma2000 Spread Spectrum Systems.

[B12] 3GPP2 C.S0024-A (V. 2.0, 2005-07), cdma2000 High Rate Packet Data Air Interface Specification.

[B13] 3GPP2 C.S0063-A (V. 1.0, 2006-04), cdma2000 High Rate Packet Data Supplemental Services.

[B14] 3GPP2 S.R0087, cdma2000 – WLAN Interworking.

[B15] 3GPP2 X.S0028-100 (V. 1.0, 2006-08), Wireless Local Area Network (WLAN) Interworking, Access to internet.

[B16] 3GPP2 X.S0028-200 (V. 1.0), Wireless Local Area Network (WLAN) Interworking, Access to Operator Services and Mobility.

[B17] IEEE 802.1ag™-2007, IEEE Standard for Local and Metropolitan Area Networks —Virtual Bridged Local Area Networks —Amendment 5: Connectivity Fault Management.

[B18] IEEE P802.21™ (Document 21-04-0087-12-0000, 2004-09), Media Independent Handover Service - Draft Technical Requirements.

[B19]IEEE P802.16g™-2007, IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 3: Management Plane Procedures and Services.

[B20]IEEE Std 802.3™-2005, Information Technology- Telecommunications and information exchange between system-Local and metropolitan area networks-Specific Requirements-Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

[B21]IETF RFC 3753 (2004-06), Mobility Related Terminology.

[B22]IETF RFC 4291 (2006-02), IP Version 6 Addressing Architecture.

[B23]IETF RFC 4443 (2006-03), Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.

[B24]IETF Internet Draft (draft-ietf-geopriv-radius-lo-10.txt, 2006-09), Carrying Location Objects in RADIUS.

[B25]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).

[B26]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).

[B27]ITU-T Recommendation Y.1541, Network performance objectives for IP-based services.

[B28]ITU-T Recommendation Z.120 (2004), Programming Languages-Formal Description Techniques (FDT)-Message Sequence Chart (MSC).

[B29]W3C Recommendation, OWL Web Ontology Language Reference.

[B30]W3C Recommendation, RDF Vocabulary Description Language 1.0: RDF Schema.

[B31]W3C Recommendation, SPARQL Query Results XML Format.

Annex B Type identifiers for information elements

(normative)

Table B-1 lists the Type identifier values for different individual IEs and IE containers.

Table B-1—Type identifiers for information elements

Type	Identifier
IE_NETWORK_TYPE	0x10000000
IE_OPERATOR_ID	0x10000001
IE_SERVICE_PROVIDER_ID	0x10000002
IE_COUNTRY_CODE	0x10000003
IE_NETWORK_ID	0x10000100
IE_NETWORK_AUX_ID	0x10000101
IE_ROAMING_PARTNERS	0x10000102
IE_COST	0x10000103
IE_NETWORK_QOS	0x10000105
IE_NETWORK_DATA_RATE	0x10000106
IE_NET_REGULAT_DOMAIN	0x10000107
IE_NET_FREQUENCY_BANDS	0x10000108
IE_NET_IP_CFG_METHODS	0x10000109
IE_NET_CAPABILITIES	0x1000010A
IE_NET_SUPPORTED_LCP	0x1000010B
IE_NET_MOB_MGMT_PROT	0x1000010C
IE_NET_EMSERV_PROXY	0x1000010D
IE_NET_IMS_PROXY_CSCF	0x1000010E
IE_NET_MOBILE_NETWORK	0x1000010F
IE_POA_LINK_ADDR	0x10000200
IE_POA_LOCATION	0x10000201
IE_POA_CHANNEL_RANGE	0x10000202
IE_POA_SYSTEM_INFO	0x10000203
IE_POA_SUBNET_INFO	0x10000204
IE_POA_IP_ADDR	0x10000205
IE_CONTAINER_LIST_OF_NETWORKS	0x10000300
IE_CONTAINER_NETWORK	0x10000301
IE_CONTAINER_POA	0x10000302

Annex C Data type definition

(normative)

C.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.

C.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 C-1 describes the encoding rules used when the data types are carried in MIH protocol messages.

Table C-1—Basic data types

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 must 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 octets 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 2's complement.
LIST(DATATYPE)	A list of values of DATATYPE	See C.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.

Table C-1—Basic data types

Type name	Definition	Binary encoding rule
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 2nd 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 Figures C-1 (LINK_ID is defined in C.3.4):

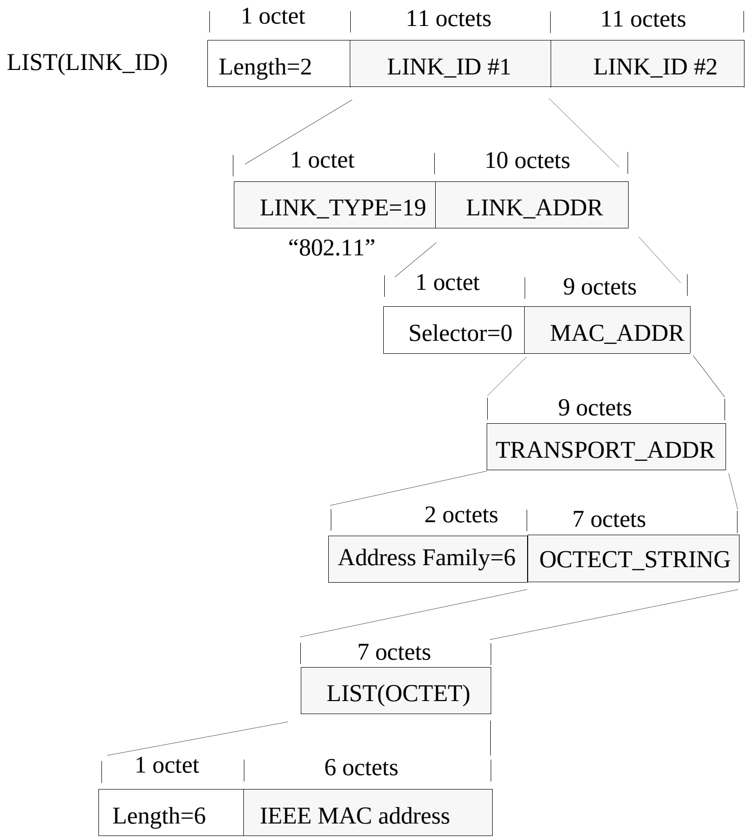


Figure C-1—Encoding example of a LIST with two LINK_ID elements

C.3 Derived data types

C.3.1 General

Derived data types are those that are derived from other data types or parent data types. A derived data type uses the same encoding as the parent data type.

C.3.2 General data types

The derived data types defined in this subclause are for general purpose only.

Table C-2—General data types

Type name	Derived from	Definition
ENUMERATED	UNSIGNED_INT(1)	An enumerated attribute. Valid Range: 0..255
BOOLEAN	ENUMERATED	Represents a boolean. 0: FALSE 1: TRUE
OCTET_STRING	LIST(OCTET)	An array of arbitrary length octets. The default encoding format is UTF-8. If a data type derived from OCTET_STRING uses other encoding format(s), the encoding format(s) must be specified in the definition of such a data type.
PERCENTAGE	UNSIGNED_INT(1)	Represents a percentage. Valid Range: 0..100
STATUS	ENUMERATED	The status of a primitive execution. 0: Success 1: Unspecified Failure 2: Rejected 3: Authorization Failure 4: Network Error

C.3.3 Data types for addresses

The data types defined in this subclause are related to addresses of network elements.

Table C-3—Data types for address

Type name	Derived from	Definition
3GPP_2G_CELL_ID	SEQUENCE(PLMN_ID, LAC, CI)	A data type to represent 3GPP 2G cell identifier
3GPP_3G_CELL_ID	SEQUENCE(PLMN_ID, CELL_ID)	A data type to represent 3GPP 3G cell identifier.
3GPP_ADDR	OCTET_STRING	A data type to represent a 3GPP transport address.
3GPP2_ADDR	OCTET_STRING	A data type to represent a 3GPP2 transport address.

Table C-3—Data types for address

Type name	Derived from	Definition
CELL_ID	UNSIGNED_INT(4)	This data type identifies a cell uniquely within 3GPP UTRAN and consists of radio network controller (RNC)-ID and C-ID as defined in 3GPP TS 25.401. Valid Range: 0..268435455
CI	OCTET(2)	The BSS and cell within the BSS are identified by Cell Identity (CI). See 3GPP TS 23.003.
IP_ADDR	TRANSPORT_ADDR	Represents an IP address. The Address Type is either 1 (IPv4) or 2 (IPv6).
LAC	OCTET(2)	Location Area Code (LAC) is a fixed length code (of 2 octets) identifying a location area within a public land mobile network (PLMN). See 3GPP TS 23.003.
LINK_ADDR	CHOICE(MAC_ADDR, 3GPP_3G_CELL_ID, 3GPP_2G_CELL_ID, 3GPP_ADDR, 3GPP2_ADDR, OTHER_L2_ADDR)	A data type to represent an address of any link-layer.
MAC_ADDR	TRANSPORT_ADDR	Represents a MAC address. The Address Type contains the one used for a specific link layer.
OTHER_L2_ADDR	OCTET_STRING	A data type to represent a link-layer address other than the address already defined. For example, SSID.
PLMN_ID	OCTET(3)	The public land mobile network (PLMN) unique identifier. PLMN_ID consists of Mobile Country Code (MCC) and Mobile Network Code (MNC). This is to represent the access network identifier. Coding of PLMN_ID is defined in 3GPP TS 25.413.
TRANSPORT_ADDR	SEQUENCE(UNSIGNED_INT(2), OCTET_STRING)	A type to represent a transport address. The UNSIGNED_INT(2) is the address type defined in http://www.iana.org/assignments/address-family-numbers.

C.3.4 Data types for link identification and manipulation

The data types defined in this subclause are used for representing attributes for identification and manipulation of links.

Table C-4—Data types for links

Type name	Derived from	Definition
BATT_LEVEL	INTEGER(1)	Represents percentage of battery charge remaining. Valid Range: -1..100. -1 indicates battery level unknown.
CHANNEL_ID	UNSIGNED_INT(2)	Channel identifier as defined in the specific link technology (e.g. standards development organization (SDO)). Valid Range: 0..65535
CONFIG_STATUS	BOOLEAN	The status of link parameter configuration. TRUE: Success FALSE: Error
DEVICE_INFO	OCTET_STRING	A non-NULL terminated string whose length shall not exceed 253 octets, representing information on manufacturer, model number, revision number of the software/firmware and serial number in displayable text.
DEV_STATES_REQ	BITMAP(16)	A list of device status request. Bitmap Values: Bit 0: DEVICE_INFO Bit 1: BATT_LEVEL Bit 2-15: (Reserved)
DEV_STATES_RSP	CHOICE(DEVICE_INFO, BATT_LEVEL)	Represents a device status.
LINK_AC_EX_TIME	UNSIGNED_INT(2)	Time (in ms) to elapse before an action needs to be taken. A value of 0 indicates that the action will be taken immediately. Time elapsed will be calculated from the instance the command arrives until the time when the execution of the action is carried out. Valid Range: 0..65535
LINK_AC_RESULT	ENUMERATED	Link action result. 0: Success 1: Failure 2: Refused 3: Incapable

Table C-4—Data types for links

Type name	Derived from	Definition
LINK_ACTION	SEQUENCE(LINK_AC_TYPE, LINK_AC_ATTR)	Link action.
LINK_AC_ATTR	BITMAP(8)	Link action attribute that can be executed along with a valid link action. Detail description of each attribute is in Table C-6. Bitmap Values: Bit 0: LINK_SCAN Bit 1: LINK_RES_RETAIN Bit 2: DATA_FWD_REQ Bit 3-7: (Reserved)
LINK_ACTION_REQ	SEQUENCE(LINK_ID, CHOICE(NULL, LINK_ADDR), LINK_ACTION, LINK_AC_EX_TIME)	A set of handover action request parameters. The choice of LINK_ADDR is to provide PoA address information when the LINK_ACTION contains the attribute for DATA_FWD_REQ.
LINK_ACTION_RSP	SEQUENCE(LINK_ID, LINK_AC_RESULT, CHOICE(NULL, LIST(LINK_SCAN_RSP)))	A set of link action returned results.
LINK_AC_TYPE	UNSIGNED_INT(1)	An action for a link. The meaning of each link action is defined in Table C-5. 0: NONE 1: LINK_DISCONNECT 2: LINK_LOW_POWER 3: LINK_POWER_DOWN 4: LINK_POWER_UP 5-255: (Reserved)
LINK_CMD_LIST	BITMAP(32)	A list of link commands. Bitmap Values: Bit 0: Reserved Bit 1: Link_Event_Subscribe Bit 2: Link_Event_Unsubscribe Bit 3: Link_Get_Parameters Bit 4: Link_Configure_Thresholds Bit 5: Link_Action Bit 6-31: (Reserved)

Table C-4—Data types for links

Type name	Derived from	Definition
LINK_CFG_PARAM	SEQUENCE(LINK_PARAM_TYPE, CHOICE(NULL, TIMER_INTERVAL), TH_ACTION, LIST(THRESHOLD))	A link configuration parameter. TH_ACTION indicates what action to apply to the listed thresholds. When “Cancel thresholds” is selected and no thresholds are specified, then all currently configured thresholds for the given LINK_PARAM_TYPE are cancelled. When “Cancel thresholds” is selected and thresholds are specified only those configured thresholds for the given LINK_PARAM_TYPE and whose threshold value match what was specified are cancelled. With “Set one-shot thresholds” the listed thresholds are first set and then each of the threshold is cancelled as soon as it is crossed for the first time.
LINK_CFG_STATUS	SEQUENCE(LINK_PARAM_TYPE, THRESHOLD, CONFIG_STATUS)	The status of link parameter configuration for each threshold specified in the THRESHOLD.
LINK_DESC_REQ	BITMAP(16)	A set of link descriptors. Bitmap Values: Bit 0: Number of Classes of Service Supported Bit 1: Number of Queues Supported Bits 2-15: (Reserved)
LINK_DESC_RSP	CHOICE(NUM_COS, NUM_QUEUE)	Descriptors of a link.
LINK_DATA_RATE	UNSIGNED_INT(4)	A type to represent the maximum data rate in kb/s. Valid Range: 0 - 2^32-1
LINK_DN_REASON	UNSIGNED_INT(1)	Represents the reason of a link down event. See Table C-7 for the enumeration values.

Table C-4—Data types for links

Type name	Derived from	Definition
LINK_EVENT_LIST	BITMAP(32)	A list of link events. The specified event is selected if the corresponding bit is set to 1. Bitmap values: Bit 0: Link_Detected Bit 1: Link_Up Bit 2: Link_Down Bit 3: Link_Parameters_Report Bit 4: Link_Going_Down Bit 5: Link_Handover_Imminent Bit 6: Link_Handover_Complete Bit 7: Link_PDU_Transmit_Status Bit 8-31: (Reserved)
LINK_GD_REASON	UNSIGNED_INT(1)	Represents the reason of a link going down. See Table C-8 for the enumeration values.
LINK_ID	SEQUENCE(LINK_TYPE LINK_ADDR)	The identifier of a link that is not associated with the peer node. The LINK_ADDR contains the address of this link.
LINK_MIHCAP_FLAG	BITMAP(8)	Represents if MIH capability is supported or not. If the bit is set, it indicates that the capability is supported. Bitmap values: Bit 1: event service (ES) supported Bit 2: command service (CS) supported Bit 3: information service (IS) supported Bit 0, 4-7: (Reserved)
LINK_PARAM	SEQUENCE(LINK_PARAM_TYPE, CHOICE(LINK_PARAM_VAL, QOS_PARAM_VAL))	Represents a link parameter type and value pair.
LINK_PARAM_802_11	UNSIGNED_INT(1)	A type to represent a link parameter for 802.11. 0: RSSI of the beacon channel, as defined in IEEE Std 802.11-2007. (This is applicable only for an MN.) 1: No QoS resource available. The corresponding LINK_PARAM_VAL is BOOLEAN set to TRUE when no QoS resources available. (This applicable when the traffic stream to be transmitted is on an access category configured for mandatory admission control and the request for bandwidth was denied by the available APs in the access network). 2: Multicast packet loss rate. 3-255: (Reserved)

Table C-4—Data types for links

Type name	Derived from	Definition
LINK_PARAM_802_16	UNSIGNED_INT(1)	A type to represent a link parameter for 802.16. 0-255: (Reserved)
LINK_PARAM_802_20	UNSIGNED_INT(1)	A type to represent a link parameter for 802.20. 0-255: (Reserved)
LINK_PARAM_802_22	UNSIGNED_INT(1)	A type to represent a link parameter for 802.22. 0-255: (Reserved)
LINK_PARAM_C2K	UNSIGNED_INT(1)	A type to represent a link parameter for CDMA2000. 0: PILOT_STRENGTH 1-255: (Reserved)
LINK_PARAM_HRPD	UNSIGNED_INT(1)	A type to represent a link parameter for CDMA2000 HRPD. 0: PILOT_STRENGTH 1-255: (Reserved)
LINK_PARAM_EDGE	UNSIGNED_INT(1)	A type to represent a link parameter for EDGE. 0-255: (Reserved)
LINK_PARAM_ETH	UNSIGNED_INT(1)	A type to represent a link parameter for Ethernet. 0-255: (Reserved)
LINK_PARAM_GEN	UNSIGNED_INT(1)	A type to represent a generic link parameter that is applicable to any link type. 0: Data Rate - the parameter value is represented as a DATA_RATE. 1: Signal Strength - the parameter value is represented as a SIG_STRENGTH. 2: Signal over interference plus noise ratio (SINR) - the parameter value is represented as an UNSIGNED_INT(2). 3:Throughput (the number of bits successfully received divided by the time it took to transmit them over the medium) - the parameter value is represented as an UNSIGNED_INT(2). 4: Packet Error Rate (representing the ratio between the number of frames received in error and the total number of frames transmitted in a link population of interest) - the parameter value is represented as a PERCENTAGE. 5-255: (Reserved)

Table C-4—Data types for links

Type name	Derived from	Definition
LINK_PARAM_GG	UNSIGNED_INT(1)	A type to represent a link parameter for GSM and GPRS. See 3GPP TS 25.008. 0: RxQual 1: RsLev 2: Mean BEP 3: StDev BEP 4-255: (Reserved)
LINK_PARAM_QOS	UNSIGNED_INT(1)	A type to represent QOS_LIST parameters. 0: Maximum number of differentiable classes of service supported. 1: Minimum packet transfer delay for all CoS, the minimum delay over a class population of interest. 2: Average packet transfer delay for all CoS, the arithmetic mean of the delay over a class population of interest. 3: Maximum packet transfer delay for all CoS, the maximum delay over a class population of interest. 4: Packet transfer delay jitter for all CoS, the standard deviation of the delay over a class population of interest. 5: Packet loss rate for all Cos, 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. 6-255: (Reserved)
LINK_PARAM_RPT	SEQUENCE(LINK_PARAM, CHOICE(NULL, THRESHOLD))	Represents a link parameter report. Includes an option of the THRESHOLD that was crossed. If no THRESHOLD is included, then this is a periodic report.
LINK_PARAM_TYPE	CHOICE(LINK_PARAM_GEN, LINK_PARAM_QOS, LINK_PARAM_GG, LINK_PARAM_EDGE, LINK_PARAM_ETH, LINK_PARAM_802_11, LINK_PARAM_C2K, LINK_PARAM_FDD, LINK_PARAM_HRPD, LINK_PARAM_802_16, LINK_PARAM_802_20, LINK_PARAM_802_22)	Measurable link parameter for which thresholds are being set.

Table C-4—Data types for links

Type name	Derived from	Definition
LINK_PARAM_VAL	UNSIGNED_INT(2)	The current value of the parameter. The format of the media-dependent value is defined in the respective media specification standard and the equivalent number of bits (i.e. first bits) of this data type is used. In case that there are remaining unused bits in the data type, these are marked as all-zeros (‘0’). Valid Range: 0..65535
LINK_PARAM_FDD	UNSIGNED_INT(1)	A type to represent a link parameter for UMTS. See 3GPP TS 25.215. 0: CPICH RSCP 1: PCCPCH RSCP 2: UTRA carrier RSSI 3: GSM carrier RSSI 4: CPICH Ec/No 5: Transport channel BLER 6: user equipment (UE) transmitted power 7-255: (Reserved)
LINK_POA_LIST	SEQUENCE(LINK_ID, LIST(LINK_ADDR))	A list of PoAs for a particular link. The LIST(LINK_ADDR) is a list of PoA link addresses and is sorted from most preferred first to least preferred last.
LINK_RR_STATUS	BOOLEAN	Represent a status of resource. TRUE: Retain resource. FALSE: Release resource.
LINK_RES_STATUS	BOOLEAN	Indicates if a resource is available or not. TRUE: Available FALSE: Not available.
LINK_SCAN_RSP	SEQUENCE(LINK_ADDR, NETWORK_ID, SIG_STRENGTH)	Represents a scan response. The LINK_ADDR contains the PoA link address. The PoA belongs to the NETWORK_ID with the given SIG_STRENGTH.
LINK_STATES_REQ	BITMAP(16)	Link states to be requested. Bit 0: OP_MODE Bit 1: CHANNEL_ID Bit 2-15: (Reserved)
LINK_STATES_RSP	CHOICE(OP_MODE,CHANNEL_ID)	The operation mode or the channel ID of the link.
LINK_STATUS_REQ	SEQUENCE(LINK_STATES_REQ, LIST(LINK_PARAM_TYPE), LINK_DESC_REQ)	Represents the possible information to request from a link.

Table C-4—Data types for links

Type name	Derived from	Definition
LINK_STATUS_RSP	SEQUENCE(LIST(LINK_STATES_RSP), LIST(LINK_PARAM), LIST(LINK_DESC_RSP))	A set of link status parameter values correspond to the LINK_STATUS_REQ.
LINK_TUPLE_ID	SEQUENCE(LINK_ID, CHOICE(NULL, LINK_ADDR))	The identifier of a link that is associated with a PoA. The LINK_ID contains the MN LINK_ADDR. The optional LINK_ADDR contains a link address of PoA.
LINK_TYPE	UNSIGNED_INT(1)	Represents the link type. Note, the values defined are made consistent with RADIUS network access server (NAS)-Port-Type definitions as specified by Internet Assigned Numbers Authority (IANA). (see IETF RFC 2865) Number assignments: 0: Reserved 1: Wireless - GSM 2: Wireless - GPRS 3: Wireless - EDGE 15: Ethernet 18: Wireless - Other 19: Wireless - IEEE 802.11 22: Wireless - CDMA2000 23: Wireless - UMTS 24: Wireless - cdma2000-HRPD 27: Wireless - IEEE 802.16 28: Wireless - IEEE 802.20 29: Wireless - IEEE 802.22
NUM_COS	UNSIGNED_INT(1)	The maximum number of differentiable classes of service supported. Valid Range: 0..255
NUM_QUEUE	UNSIGNED_INT(1)	The number of transmit queues supported. Valid Range: 0..255
OP_MODE	UNSIGNED_INT(1)	The link power mode. 0: Normal Mode 1: Power Saving Mode 2: Powered Down 3-255: (Reserved)
SIG_STRENGTH	PERCENTAGE	Represents the percentage of receiver's strongest signal strength in dBm.
TH_ACTION	ENUMERATED	0: Set normal threshold 1: Set one-shot threshold 2: Cancel threshold

Table C-4—Data types for links

Type name	Derived from	Definition
THRESHOLD	SEQUENCE(THRESHOLD_VAL, THRESHOLD_X_DIR)	A link threshold. The threshold is considered crossed when the value of the link parameter passes the threshold in the specified direction.
THRESHOLD_VAL	UNSIGNED_INT(2)	Threshold value. The format of the media-dependent value is defined in the respective media specification standard and the equivalent number of bits (i.e. first bits) of this data type is used. In case that there are remaining unused bits in the data type, these are marked as all-zeros (‘0’). Valid Range: 0..65535
THRESHOLD_X_DIR	UNSIGNED_INT(1)	The direction the threshold is to be crossed. 0: ABOVE_THRESHOLD 1: BELOW_THRESHOLD 2-255: (Reserved)
TIMER_INTERVAL	UNSIGNED_INT(2)	This timer value (ms) is used to set the interval between periodic reports. Valid Range: 0..65535

Table C-5—Link actions

Action name	Description
LINK_DISCONNECT	Disconnect the link connection directly.
LINK_LOW_POWER	Cause the link to adjust its battery power level to be low power consumption.
LINK_POWER_DOWN	Cause the link to power down and turn off the radio.
LINK_POWER_UP	Cause the link to power up and establish L2 connectivity. For UMTS link type, power up lower layers and establish PDP context.

Table C-6—Link action attributes

Action name	Description
DATA_FWD_REQ	This indication requires the buffered data at the old serving PoA entity to be forwarded to the new target PoA entity in order to avoid data loss. This action can be taken immediately after the old serving PoS receives MIH_N2N_HO_Commit response message from the new target PoS, or the old serving PoS receives MIH_Net_HO_Commit response message from the MN. This is not valid on UMTS link type.
LINK_RES_RETAIN	The link will be disconnected but the resource for the link connection still remains so reestablishing the link connection later can be more efficient.
LINK_SCAN	Cause the link to perform a scan.

Table C-7—Link down reason code

Reason code	Reason	Description
0	Explicit disconnect	The link is down because of explicit disconnect procedures initiated either by MN or network.
1	Packet timeout	The link is down because no acknowledgements were received for transmitted packets within the specified time limit.
2	No resource	The link is down because there were no resources to maintain the connection.
3	No broadcast	The link is down because broadcast messages (such as beacons in 802.11 management frames) could not be received by MN.
4	Authentication failure	Authentication failure.
5	Billing failure	Billing failure.
6-127	(*Reserved*)	Reserved for IEEE 802.21 future use.
128-255	Vendor specific reason codes	Vendors specify their own specific reason codes in this range.

Table C-8—Link going down reason code

Reason code	Reason	Description
0	Explicit disconnect	The link is going to be down because explicit disconnect procedures will be initiated either by MN or network. For example, when a BS has decided to shutdown for administrative reasons or an operator of the terminal has decided to execute a handover manually, a Link_Going_Down trigger is sent to the MIHF.
1	Link parameter degrading	The link is going to be down because broadcast messages (such as beacons in 802.11 management frames) could not be received by MN.
2	Low power	The link is going to be down because the power level of the terminal is low and the current link will not be maintained in such a low power level. Mobile terminals usually have limited battery supply, and when the battery level of the terminal is low, a terminal can choose a link that has lower power consumption for handover according to the received Link_Going_Down triggers with the this reason code. This will lengthen the usable time for the terminal.
3	No resource	The link is going to be down because there will be no resources to maintain the current connection. For example, a BS that has too many users can send Link_Going_Down indications to terminals when the links with them can not be kept because of insufficient resources. Another example is that users with higher priority can preempt the ones with lower priority when no more resources can be allocated in 3GPP, and this can also cause a Link_Going_Down indication with this reason code.
4-127	(*Reserved*)	Reserved for IEEE 802.21 future use.
128-255	Vendor specific reason codes	Vendors specify their own specific reason codes in this range.

C.3.5 Data types for QoS

The data types defined in this subclause are related to QoS.

Table C-9—Data types for QoS

Type name	Derived from	Definition
QOS_LIST	SEQUENCE(NUM_COS_TYPES, LIST(MIN_PK_TX_DELAY), LIST(AVG_PK_TX_DELAY), LIST(MAX_PK_TX_DELAY), LIST(PK_DELAY_JITTER), LIST(PK_LOSS_RATE))	A list of Class of Service (CoS) parameters.
NUM_COS_TYPES	UNSIGNED_INT(1)	A type to represent the maximum number of differentiable classes of service supported. Valid Range: 0..255

Table C-9—Data types for QoS

Type name	Derived from	Definition
MIN_PK_TX_DELAY	SEQUENCE(COS_ID, UNSIGNED_INT(2))	A type to represent the minimum packet transfer delay in ms for the specific CoS specified by the COS_ID.
AVG_PK_TX_DELAY	SEQUENCE(COS_ID, UNSIGNED_INT(2))	A type to represent the average packet transfer delay in ms for the specific CoS specified by the COS_ID.
MAX_PK_TX_DELAY	SEQUENCE(COS_ID, UNSIGNED_INT(2))	A type to represent the maximum packet transfer delay in ms for the specific CoS specified by the COS_ID.
PK_DELAY_JITTER	SEQUENCE(COS_ID, UNSIGNED_INT(2))	A type to represent the packet transfer delay jitter in ms for the specific CoS specified by the COS_ID.
PK_LOSS_RATE	SEQUENCE(COS_ID, UNSIGNED_INT(2))	A type to represent the packet loss rate for the specific CoS specified by the COS_ID. The loss rate is equal to the integer part of the result of multiplying -100 times the log10 of the ratio between the number of packets lost and the total number of packets transmitted in the class population of interest.
COS_ID	UNSIGNED_INT(1)	A type to represent a class of service identifier. Valid Range: 0 - 255
QOS_PARAM_VAL	CHOICE(NUM_COS_TYPES, LIST(MIN_PK_TX_DELAY), LIST(AVG_PK_TX_DELAY), LIST(MAX_PK_TX_DELAY), LIST(PK_DELAY_JITTER), LIST(PK_LOSS_RATE))	A choice of Class of Service (CoS) parameters.

C.3.6 Data types for location

Table C-10—Data types for location

Type name	Derived from	Definition
LOCATION	CHOICE(CIVIC_LOC, GEO_LOC, CELL_ID)	A type to represent the format and value of the location information. The location can be civic location, geospacial location, or a cellular ID value as reference location.
CIVIC_LOC	CHOICE(BIN_CIVIC_LOC, XML_CIVIC_LOC)	A type to represent a civic address.

Table C-10—Data types for location

Type name	Derived from	Definition
BIN_CIVIC_LOC	SEQUENCE(CNTRY_CODE, CIVIC_ADDR)	A type to represent a binary-formatted civic address. See CNTRY_CODE and CIVIC_ADDR definitions.
XML_CIVIC_LOC	OCTET_STRING	A type to represent an XML-formatted civic location. Civic address elements, as described in IETF RFC 4119.
CIVIC_ADDR	OCTET_STRING	A type to represent civic address elements in BIN_CIVIC_LOC. Civic address elements, as described in IETF RFC 4776.
GEO_LOC	CHOICE(BIN_GEO_LOC, XML_GEO_LOC)	A type to represent a geospatial location.
BIN_GEO_LOC	OCTET(OCTET(16)	A type to represent a binary-formatted geospatial location. See Table C-11.
XML_GEO_LOC	OCTET_STRING	A type to represent an XML-formatted geospatial location. Geo address elements as described in IETF RFC 4119. For example, <gml:location> <gml:Point gml:id="point1" srsName="epsg:4326"> <gml:coordinates>37:46:30N 122:25:10W</gml:coordinates> </gml:Point> </gml:location>

Table C-11—Value field format of PoA location information (geospatial location)

Syntax	Length (bits)	Notes (See IETF RFC 3825 for details)
LatitudeResolution (LaRes)	6	Latitude resolution. six bits indicating the number of valid bits in the fixed-point value of Latitude. Any bits entered to the right of this limit should not be considered valid and might be purposely false, or zeroed by the sender.
Latitude	34	A 34 bit fixed point value consisting of nine bits of integer and 25 bits of fraction. Latitude should be normalized to within +/- 90 degrees. Positive numbers are north of the equator and negative numbers are south of the equator.
LongitudeResolution (LoRes)	6	Longitude resolution. six bits indicating the number of valid bits in the fixed-point value of Longitude. This value is the number of high-order Longitude bits that should be considered valid. Any bits entered to the right of this limit should not be considered valid and might be purposely false, or zeroed by the sender.
Longitude	34	A 34 bit fixed point value consisting of nine bits of integer and 25 bits of fraction. Longitude should be normalized to within +/- 180 degrees. Positive values are East of the prime meridian and negative (2s complement) numbers are West of the prime meridian.

Table C-11—Value field format of PoA location information (geospatial location)

Syntax	Length (bits)	Notes (See IETF RFC 3825 for details)
AltitudeType (AT)	4	Following codes are defined: 1: Meters: in 2s-complement fixed-point 22-bit integer part with 8-bit fraction. If AT = 1, an AltRes value 0.0 would indicate unknown altitude. The most precise Altitude would have an AltRes value of 30. Many values of AltRes would obscure any variation due to vertical datum differences. 2: Floors: in 2s-complement fixed-point 22-bit integer part with 8-bit fraction. AT = 2 for Floors enables representing altitude in a form more relevant in buildings that have different floor-to-floor dimensions.
AltitudeResolution (AltRes)	6	Altitude resolution. six bits indicating the number of valid bits in the altitude. Values above 30 (decimal) are undefined and reserved.
Altitude	30	A 30 bit value defined by the AT field.
Datum	8	Following codes are defined: 1: WGS 2: NAD 83 (with associated vertical datum for North American vertical datum for 1998) 3: NAD 83 (with associated vertical datum for Mean Lower Low Water (MLLW))

C.3.7 Data types for IP configuration

Table C-12—Data types for IP configuration

Type name	Derived from	Definition
IP_CFG_MTHDS	BITMAP(32)	A set of IP configuration methods. Bit 0: IPv4 static configuration Bit 1: IPv4 dynamic configuration (DHCPv4) Bit 2: Mobile IPv4 with foreign agent (FA) care-of address (CoA) (FA-CoA) Bit 3: Mobile IPv4 without FA (Co-located CoA) Bits 4-10: reserved for IPv4 address configurations Bit 11: IPv6 stateless address configuration Bit 12: IPv6 stateful address configuration (DHCPv6) Bit 13: IPv6 manual configuration Bits 14-31: (Reserved)
IP_CFG_STATUS	BITMAP(8)	Status of the IP configuration methods. Bit 0: IP configuration Method is not available Bit 1: DHCP Server address is not available Bit 2: FA address is not available Bit 3: Access Router Address is not available Bit 4: No information is provided due to accessibility of same entity (FA, Access Router, DHCP Server, etc.) Bit 5-7: (Reserved)

Table C-12—Data types for IP configuration

Type name	Derived from	Definition
IP_MOB_MGMT	BITMAP(16)	Indicates the supported mobility management protocols. Bit 0: Mobile IPv4 (IETF RFC 3344) Bit 1: Mobile IPv4 Regional Registration (IETF RFC 4857) Bit 2: Mobile IPv6 (IETF RFC 3775) Bit 3: Hierarchical Mobile IPv6 (IETF RFC 4140) Bit 4: Low Latency Handoffs (IETF RFC 4881) Bit 5: Fast Handovers for Mobile IPv6 (IETF RFC 4068) Bit 6: IKEv2 Mobility and Multihoming Protocol (IETF RFC 4555) Bit 7-15: (Reserved)
IP_PREFIX_LEN	UNSIGNED_INT(1)	The length of an IP subnet prefix. Valid Range: 0..32 for IPv4 subnet. 0..64, 65..127 for IPv6 subnet. (IETF RFC 4291)
IP_RENEWAL_FLAG	BOOLEAN	Indicates whether MN’s IP address needs to be changed or not. TRUE: Change required. FALSE: Change not required.
IP_SUBNET_INFO	SEQUENCE(IP_PREFIX_LEN, IP_ADDR)	Represent an IP subnet. The IP_PREFIX_LEN contains the bit length of the prefix of the subnet to which the IP_ADDR belongs.

C.3.8 Data types for information elements

Data types defined in this subclause are used only by IEs.

Table C-13—Data types for information elements

Type name	Derived from	Definition
NET_AUX_ID	OCTET_STRING	A type to represent an auxiliary access network identifier. This is HESSID if network type is IEEE 802.11.
NETWORK_ID	OCTET_STRING	A type to represent a network identifier. A non-NULL terminated string whose length shall not exceed 253 octets.
BAND_CLASS	UNSIGNED_INT(1)	CDMA band class.
BANDWIDTH	UNSIGNED_INT(2)	Channel bandwidth in kb/s.
BASE_ID	UNSIGNED_INT(2)	Base station identifier.
BURST_PROF	SEQUENCE(DOWN_BP, UP_BP)	Burst profile

Table C-13—Data types for information elements

Type name	Derived from	Definition
CH_RANGE	SEQUENCE(UNSIGNED_INT(4), UNSIGNED_INT(4))	A type that contains two numbers. The first unsigned integer is the low range. The second unsigned integer is the high range. Both values are in KHz. The first unsigned integer value should always be less or equal to the second unsigned integer.
COST	SEQUENCE(COST_UNIT, COST_VALUE, COST_CURR)	A type to represent a cost.
COST_CURR	OCTET(3)	A type to represent the currency of a cost. A three-letter currency code (e.g., “USD”) specified by ISO 4217.
COST_UNIT	UNSIGNED_INT(1)	A type to represent the unit of a cost. 0: second 1: minute 2: hours 3: day 4: week 5: month 6: year 7: free 8: flat rate 9-255: (Reserved)
COST_VALUE	SEQUENCE(UNSIGNED_INT(4), UNSIGNED_INT(2))	A type to represent the value of a cost. The first 4-octet contains the integer part of the cost. The last 2-octet contains the fraction part where it represents a 3-digit fraction. Therefore, the value range of the fraction part is [0,999]. For example, for a value of “0.5”, the integer part is zero and the fraction part is 500.
CNTRY_CODE	OCTET(2)	Country code, represented as two letter ISO 3166-1 country code in capital ASCII letters.
DATA_RATE	UNSIGNED_INT(4)	A type to represent the maximum data rate in kb/s. Valid Range: 0..2^32-1
DCD_UCD	SEQUENCE(BASE_ID, BANDWIDTH, DU_CTR_FREQ, EIRP, GAP, BURST_PROF, CDMA_CODES)	A type to represent DCD _UCD.
DOWN_BP	BITMAP(256)	A List of FEC Code Type for Downlink burst. Refer to 11.4.1 in IEEE 802.16Rev2/D4.0 [ref].
EIRP	INTEGER(1)	BS’s effective isotropic radiated power level. Signed in units of 1 dBm.

Table C-13—Data types for information elements

Type name	Derived from	Definition
FQDN	OCTET_STRING	The fully qualified domain name of a host as described in IETF RFC 2181.
DU_CTR_FREQ	INTEGER(8)	Downlink/Uplink center frequency in KHz.
FREQ_BANDS	LIST(UNSIGNED_INT(4))	A list of frequency bands. The values are in KHz.
FREQ_ID	INTEGER(2)	Identifier the carrier frequency. Valid Range: 0..65535
FQ_CODE_NUM	INTEGER(2)	UMTS scrambling code, cdma2000 Walsh code. Valid Range: 0..65535
GAP	SEQUENCE(TTG, RTG)	
HO_CODE	INTEGER(1)	HANDOVER_RANGING_CODE. Refer to 11.3.1 in IEEE 802.16Rev2/D4.0 [ref].
INIT_CODE	INTEGER(1)	INITIAL_RANGING_CODE. Refer to 11.3.1 in IEEE 802.16Rev2/D4.0 [ref].
IP4_ADDR	OCTET(4)	An IPv4 address as described in IETF RFC 791.
IP6_ADDR	OCTET(16)	An IPv6 address as described in IETF RFC 2373.
IP_CONFIG	SEQUENCE(IP_CFG_MTHDS, CHOICE(NULL, DHCP_SERV), CHOICE(NULL, FN_AGNT), CHOICE(NULL, ACC_RTR))	IP Configuration Methods supported by the access network.

Table C-13—Data types for information elements

Type name	Derived from	Definition
NET_CAPS	BITMAP(32)	These bits provide high level capabilities supported on a network. Bitmap Values: Bit 0: Security – Indicates that some level of security is supported when set. Bit 1: QoS Class 0 – Indicates that QoS for class 0 is supported when set. * Bit 2: QoS Class 1 – Indicates that QoS for class 1 is supported when set. * Bit 3: QoS Class 2 – Indicates that QoS for class 2 is supported when set; Otherwise, no QoS for class 2 support is available. Bit 4: QoS Class 3 – Indicates that QoS for class 3 is supported when set; Otherwise, no QoS for class 3 support is available. Bit 5: QoS Class 4 – Indicates that QoS for class 4 is supported when set; Otherwise, no QoS for class 4 support is available. Bit 6: QoS Class 5 – Indicates that QoS for class 5 is supported when set; Otherwise, no QoS for class 5 support is available. Bit 7: Internet Access – Indicates that Internet access is supported when set; Otherwise, no Internet access support is available. Bit 8: Emergency Services – Indicates that some level of emergency services is supported when set; Otherwise, no emergency service support is available. Bit 9: MIH Capability – Indicates that MIH is supported when set; Otherwise, no MIH support is available. Bit 10-31: (Reserved) * Note that the definitions of the QOS classes are according to ITU Y.1541.[ref]
NETWORK_TYPE	SEQUENCE(CHOICE(NULL, LINK_TYPE), CHOICE(NULL, REVISION), CHOICE(NULL, TYPE_EXT))	A type to represent a network type and its revision. See Table C-14 for details.
OPERATOR_ID	SEQUENCE(OP_NAME, OP_NAMESPACE)	A type to represent an operator identifier.
OP_NAME	OCTET_STRING	A type to represent an operator name. The value uniquely identifies the operator name within the scope of the OP_NAMESPACE. The value is a non NULL terminated string whose length shall not exceed 253 octets.

Table C-13—Data types for information elements

Type name	Derived from	Definition
OP_NAMESPACE	UNSIGNED_INT(1)	A type to represent a type of operator name. 0: GSM/UMTS 1: CDMA 2: REALM (as defined in [30]). 3: ITU-T/TSB 4: General 5-255: (Reserved)
PARAMETERS	CHOICE(DCD_UCD, SIB, SYS_PARAMS)	A data type to represent system information depending on the network type. DCD_UCD: IEEE 802.16 SIB: UMTS SYS_PARAMS: cdma2000
PILOT_PN	INTEGER(2)	Pilot PN sequence offset index.
PROXY_ADDR	CHOICE(IP4_ADDR, IP6_ADDR, FQDN)	L3 address of a proxy server.
CDMA_CODES	SEQUENCE(INIT_CODE, HO_CODE)	A set of CDMA ranging codes.
REGU_DOMAIN	SEQUENCE(CNTRY_CODE, UNSIGNED_INT(1))	A type to represent a regulatory domain. A regulatory domain is identified by a country code (CNTRY_CODE) and a regulatory class (UNSIGNED_INT(1)). The regulatory class values are defined in Annex J of IEEE P802.11k/D13.0, (2008-03) draft specification.
REVISION	BITMAP(64)	A network type revision. See Table C-14.
ROAMING_PTNS	LIST(OPERATOR_ID)	A list of roaming partners.
RTG	INTEGER(1)	Receive transition gap in microseconds. Physical slot unit.
SP_ID	OCTET_STRING	A service provider identifier. A non-NULL terminated string whose length shall not exceed 253 octets.
SIB	SEQUENCE(CELL_ID, FQ_CODE_NUM)	A type to represent UMTS system information block (SIB).

Table C-13—Data types for information elements

Type name	Derived from	Definition
SUPPORTED_LCP	UNSIGNED_INT(1)	A type represent supported Location Configuration Protocol. (LCP). Location by Reference (LbyR). Values represent LCPs: 0: NULL 1: LLDP 2: LbyR with LLDP 3-10: (Reserved) 11: LLDP-MED 12: LbyR with LLDP-MED 13-20: (Reserved) 21: U-TDoA 22: D-TDoA 23-30: (Reserved) 31: DHCP 32: LbyR with DHCP 33-40: (Reserved) 41: OMA SUPL 42: IEEE 802.11 LCI 43-50: (Reserved) 51: HELD 52: LbyR with HELD 53-255: (Reserved)
SYSTEM_INFO	SEQUENCE(NETWORK_TYPE, LINK_ADDR, CHOICE(NULL, PARAMETERS))	A type to represent system information.
SYS_PARAMS	SEQUENCE (BASE_ID, PILOT_PN, FREQ_ID, BAND_CLASS)	CDMA2000 system parameters.
TTG	INTEGER(2)	Transmit transition gap in microseconds. Physical slot unit.
TYPE_EXT	OCTET_STRING	A generic type extension contained indicating a flexible length and format field. The content is to be defined and filled by the appropriate SDO or service provider consortium, etc. The value is a non-NULL terminated string whose length shall not exceed 253 octets.
UP_BP	BITMAP(256)	A List of FEC Code Type for Uplink burst. Refer to 11.3.1 in IEEE 802.16Rev2/D4.0 [ref].

Table C-14—Network type and revision representation

Network	Link type	Revision
(*Reserved*)	0	N/A
Wireless - GSM	1	N/A
Wireless - GPRS	2	N/A
Wireless - EDGE	3	N/A
(*Reserved*)	4-14	N/A
Ethernet - IEEE 802.3	15	Bit 0: 10 Mb Bit 1: 100 Mb Bit 2: 1000 Mb Bit 3-63: (Reserved) The above bits represent the link speeds that Ethernet supports. The capability information of twisted pair Ethernet link can be obtained via auto-negotiation as defined in Clause 28 of the 1998 edition of IEEE Std 802.3.
(*Reserved*)	16-17	N/A
Wireless - Other	18	N/A
Wireless - IEEE 802.11	19	Bit 0: 2.4 GHz Bit 1: 5 GHz Bit 2-63 (Reserved) The above bits represent the frequency band that IEEE 802.11 link supports. The capability information of IEEE 802.11 link can further be represented as defined in 7.3.1.4 of IEEE Std 802.11-2007.
(*Reserved*)	20-21	N/A
Wireless - CDMA2000	22	N/A
Wireless - UMTS	23	Bit 0: Rel-99 Bit 1: Rel-4 Bit 2: Rel-5 (w/ HSDPA) Bit 3: Rel-6 (w/ HSUPA) Bit 4: Rel-7 (MIMO/OFDM) Bit 5: Rel-8 Bit 6-63: (Reserved)
Wireless - cdma2000-HRPD	24	Bit 0: Rev-0 Bit 1: Rev-A Bit 2: Rev-B Bit 3: Rev-C Bit 4-63: (Reserved)
(*Reserved*)	25-26	N/A
Wireless - IEEE 802.16	27	Bit 0: 2.5 GHz Bit 1: 3.5 GHz Bit 2-63: (Reserved) The above bits represent the frequency band that IEEE 802.16 link supports. The system profiles of IEEE 802.16 link can further be represented as defined in clause 12 (12.3 and 12.4) of IEEE Std 802.16-2007.

Table C-14—Network type and revision representation

Network	Link type	Revision
Wireless - IEEE 802.20	28	N/A
Wireless - IEEE 802.22	29	N/A
(*Reserved*)	30-255	N/A

NOTE- The Link type values in Table C-14 are deliberately made consistent with RADIUS network access server (NAS)-Port-Type definitions as specified by Internet Assigned Numbers Authority (IANA).

C.3.9 Data types for information service query

C.3.9.1 Binary representation

Table C-15—Data types for binary query

Type name	Derived from	Definition
CURR_PREF	COST_CURR	A type to indicate currency preference.
IE_TYPE	UNSIGNED_INT(4)	A type to represent an IE type. See Table B-1 for more information.
IQ_BIN_DATA	SEQUENCE(CHOICE(NULL, QUERIER_LOC), CHOICE(NULL, NET_TYPE_INC), CHOICE(NULL, NETWK_INC), CHOICE(NULL, RPT_TEMPL), CHOICE(NULL, RPT_LIMIT), CHOICE(NULL, CURR_PREF))	Represents a binary query. There should exist at least one of the query data type QUERIER_LOC, NET_TYPE_INC, or NETWK_INC. One CURR_PREF at most is included in an Info Query Binary TLV. If included, it indicates to the MIIS server the preferred currency the returned cost should be represented in. If the MIIS server cannot return the cost in the specified currency, it can return the cost in other currencies.
NGHB_RADIUS	UNSIGNED_INT(4)	The radius in meters from the center point of querier's location. Valid Range: 0..2^32-1
NETWK_INC	LIST(NETWORK_ID)	A type to represent a list of network identifiers.

Table C-15—Data types for binary query

Type name	Derived from	Definition
NET_TYPE_INC	BITMAP(32)	A type to represent a set of link types. The value is a four octet bitmap: Bit 0: Wireless - GSM Bit 1: Wireless - GPRS Bit 2: Wireless - EDGE Bit 3: IEEE 802.3 (Ethernet) Bit 4: Wireless - Other Bit 5: Wireless - IEEE 802.11 Bit 6: Wireless - CDMA2000 Bit 7: Wireless - UMTS Bit 8: Wireless - cdma2000-HRPD Bit 9: Wireless - IEEE 802.16 Bit 10: Wireless - IEEE 802.20 Bit 11: Wireless - IEEE 802.22 Bit 12-31: (*Reserved AND shall be always set to "0"*)
QUERIER_LOC	SEQUENCE(CHOICE(NULL, LOCATION), CHOICE(NULL, LINK_ADDR), CHOICE(NULL, NGHB_RADIUS))	A type to represent a querier's location. It is not valid to use both LOCATION and LINK_ADDR at the same time.
RPT_LIMIT	SEQUENCE(UNSIGNED_INT(2), UNSIGNED_INT(2))	A type to represent a report limitation. The first UNSIGNED_INT(2) contains the maximum number of IEs in the IR_BIN_DATA. The second UNSIGNED_INT(2) contains the starting entry number (offset = 1 points to the first entry) from which a chunk of entries are to be included in the IQ_BIN_DATA. It is assumed that the IS server generates the same ordered list of entries for queries from the same IS client with the same IR_BIN_DATA content (except for RPT_LIMIT) before the limitation on the RPT_LIMIT is applied.
RPT_TEMPL	LIST(IE_TYPE)	A type to represent a list of IE types. Inclusion of any IE type is optional.

C.3.9.2 RDF representation

Table C-16—Data type for RDF query

Type name	Derived from	Definition
IQ_RDF_SCHM	OCTET_STRING	A type to represent the URL of an RDF schema to obtain.
IQ_RDF_DATA	SEQUENCE(CHOICE(NULL, MIME_TYPE), OCTET_STRING)	Represents RDF query. If MIME_TYPE is omitted, MIME type "application/sparql-query" is used. Each OCTET_STRING is formatted with the MIME type.
MIME_TYPE	OCTET_STRING	Represents MIME type.

C.3.10 Data types for information service response

C.3.10.1 Binary representation

Table C-17—Data type for binary information query response

Type name	Derived from	Definition
IR_BIN_DATA	LIST(INFO_ELEMENT)	A type to represent a binary query response data.

C.3.10.2 RDF representation

Table C-18—Data type for RDF information query response

Type name	Derived from	Definition
IR_RDF_DATA	SEQUENCE(CHOICE(NULL, MIME_TYPE), OCTET_STRING)	Represents RDF data query result. If MIME_TYPE is omitted, MIME type “application/ sparql-results+xml” is used. OCTET_STRING is formatted with the MIME type.
IR_SCHM_URL	OCTET_STRING	An URL of an RDF schema.
IR_RDF_SCHM	SEQUENCE(CHOICE(NULL, MIME_TYPE), OCTET_STRING)	Represents an RDF schema. If MIME_TYPE is omitted, MIME type “application/xml” is used. OCTET_STRING is formatted with the MIME type.

C.3.11 Data type for MIHF identification

Table C-19—Data type for MIHF identification

Type name	Derived from	Definition
MIHF_ID	OCTET_STRING	The MIHF Identifier: MIHF_ID is a network access identifier (NAI). NAI shall be unique as per IETF RFC 4282. If MIHF entity resides in the network node then MIHF_ID is the fully qualified domain name or IP address of the entity that hosts the MIH Services. The maximum length is 253 octets.

C.3.12 Data type for MIH capabilities

Table C-20—Data type for MIH capabilities

Type name	Derived from	Definition
EVT_CFG_INFO	CHOICE(LIST(LINK_DET_CFG), LIST(LINK_CFG_PARAM))	Represents additional configuration information for event subscription. The list of LINK_DET_CFG contains additional filtering when subscribing to link detected events. The list of LINK_CFG_PARAM contains additional filtering when subscribing to link parameter report events.
LINK_DET_CFG	SEQUENCE(CHOICE(NULL, NETWORK_ID), CHOICE(NULL, SIG_STRENGTH), CHOICE(NULL, LINK_DATA_RATE))	A data type for configuring link detected event trigger.
LINK_DET_INFO	SEQUENCE (LINK_TUPLE_ID, NETWORK_ID, NET_AUX_ID, SIG_STRENGTH, UNSIGNED_INT(2), LINK_DATA_RATE, LINK_MIHCAP_FLAG, NET_CAPS)	Information of a detected link. LINK_TUPLE_ID is the link detected. NETWORK_ID is the access network identifier. NET_AUX_ID is an auxiliary access network identifier if applicable. SIG_STRENGTH is the signal strength of the detected link. UNSIGNED_INT(2) is the SINR value of the link. LINK_DATA_RATE is the maximum transmission rate on the detected link. LINK_MIHCAP_FLAG indicates which MIH capabilities are supported on the detected link. NET_CAPS is the network capability supported by the network link.
MBB_HO_SUPP	SEQUENCE(NETWORK_TYPE, NETWORK_TYPE, BOOLEAN)	Indicates if make before break is supported FROM the first network type TO the second network type. The BOOLEAN value assignment: True: Make before break is supported. False: Make before break is not supported.
MIH_CMD_LIST	BITMAP(32)	A list of MIH commands. Bitmap Values: Bit 0: MIH_Link_Get_Parameters Bit 1: MIH_Link_Configure_Thresholds Bit 2: MIH_Link_Actions Bit 3: MIH_Net_HO_Candidate_Query MIH_Net_HO_Commit MIH_N2N_HO_Query_Resources MIH_N2N_HO_Commit MIH_N2N_HO_Complete Bit 4: MIH_MN_HO_Candidate_Query MIH_MN_HO_Commit MIH_MN_HO_Complete Bit 5-31: (Reserved)

Table C-20—Data type for MIH capabilities

Type name	Derived from	Definition
MIH_EVT_LIST	BITMAP(32)	A list of MIH events. Bitmap Values: Bit 0: MIH_Link_Detected Bit 1: MIH_Link_Up Bit 2: MIH_Link_Down Bit 3: MIH_Link_Parameters_Report Bit 4: MIH_Link_Going_Down Bit 5: MIH_Link_Handover_Imminent Bit 6: MIH_Link_Handover_Complete Bit 7: MIH_Link_PDU_Transmit_Status Bit 8-31: (Reserved)
MIH_IQ_TYPE_LST	BITMAP(64)	A list of IS query types. Bitmap Values: Bit 0: Binary data Bit 1: RDF data Bit 2: RDF schema URL Bit 3: RDF schema Bit 4: IE_NETWORK_TYPE Bit 5: IE_OPERATOR_ID Bit 6: IE_SERVICE_PROVIDER_ID Bit 7: IE_COUNTRY_CODE Bit 8: IE_NETWORK_ID Bit 9: IE_NETWORK_AUX_ID Bit 10: IE_ROAMING_PARTNERS Bit 11: IE_COST Bit 12: IE_NETWORK_QOS Bit 13: IE_NETWORK_DATA_RATE Bit 14: IE_NET_REGULT_DOMAIN Bit 15: IE_NET_FREQUENCY_BANDS Bit 16: IE_NET_IP_CFG_METHODS Bit 17: IE_NET_CAPABILITIES Bit 18: IE_NET_SUPPORTED_LCP Bit 19: IE_NET_MOB_MGMT_PROT Bit 20: IE_NET_EMSERV_PROXY Bit 21: IE_NET_IMS_PROXY_CSCF Bit 22: IE_NET_MOBILE_NETWORK Bit 23: IE_POA_LINK_ADDR Bit 24: IE_POA_LOCATION Bit 25: IE_POA_CHANNEL_RANGE Bit 26: IE_POA_SYSTEM_INFO Bit 27: IE_POA_SUBNET_INFO Bit 28: IE_POA_IP_ADDR Bit 29- 63: (Reserved)

Table C-20—Data type for MIH capabilities

Type name	Derived from	Definition
MIH_TRANS_LST	BITMAP(16)	A list of supported transports. Bitmap Values: Bit 0: UDP Bit 1: TCP Bit 2-15: (Reserved)
NET_TYPE_ADDR	SEQUENCE(NETWORK_TYPE, LINK_ADDR)	Represent a link address of a specific network type.

C.3.13 Data type for MIH registration

Table C-21—Data type for MIH registration

Type name	Derived from	Definition
REG_REQUEST_CODE	ENUMERATED	The registration code: 0 - Registration 1 - Re-Registration

C.3.14 Data types for handover operation

Table C-22—Data type for handover operation

Type name	Derived from	Definition
ASGN_RES_SET	SEQUENCE (QOS_ LIST, TSP_CONTAINER)	Set of resource parameters reserved and assigned by the target network to the MN for performing handover to a network PoA. The transparent container is from target to source, which includes the required configuration of the reserved resources at the target network.
HO_CAUSE	UNSIGNED_INT(1)	Represents the reason for performing a handover. Same enumeration list as link down reason code. See Table C-7.
HO_RESULT	ENUMERATED	Handover result. 0: Success 1: Failure 2: Rejected
HO_STATUS	ENUMERATED	Represents the permission for handover. 0: HandoverPermitted 1: HandoverDeclined
PREDEF_CFG_ID	INTEGER(1)	Pre-defined configuration identifier. 0..255

Table C-22—Data type for handover operation

Type name	Derived from	Definition
RQ_RESULT	SEQUENCE(LINK_POA_LIST, QOS_LIST, CHOICE(NULL, IP_CFG_MTHDS), CHOICE(NULL, DHCP_SERV), CHOICE(NULL, FN_AGNT), CHOICE(NULL, ACC_RTR))	Represents the result of network resource query. The LINK_POA_LIST is a list of potential PoAs for a given link with the same IP configuration information. The list is sorted from most preferred first to least preferred last. The QOS_LIST contains the available resources for this list of PoAs. IP_CFG_MTHDS represents the IP configuration methods applicable.
DHCP_SERV	IP_ADDR	IP address of candidate DHCP Server. It is only included when dynamic address configuration is supported.
FN_AGNT	IP_ADDR	IP address of candidate Foreign Agent. It is only included when Mobile IPv4 is supported.
ACC_RTR	IP_ADDR	IP address of candidate Access Router. It is only included when IPv6 Stateless configuration is supported.
REQ_RES_SET	SEQUENCE (QOS_LIST, TSP_CONTAINER, HO_CAUSE)	Set of resource parameters required for performing admission control and resource reservation for the MN at the target network. The transparent container is from source to target, which includes the required MN configuration for admitting the new connection at the target network and reserving resources.
TGT_NET_INFO	CHOICE (SEQUENCE(NETWORK_ID, CHOICE(NULL, NET_AUX_ID)), LINK_ADDR)	Represents the handover commit information. LINK_ADDR is the target PoA link address.
TSP_CARRIER	OCTET_STRING	Transparent carrier containing link specific information whose content and format are to be specified by the link specific SDO.
TSP_CONTAINER	CHOICE (NULL, PREDEF_CFG_ID, TSP_CARRIER)	Transparent container. If the value is null, this parameters is not available.

C.3.15 Data types for MIH_NET_SAP primitives

Table C-23—Data type for MIH_NET_SAP primitives

Type name	Derived from	Definition
TRANSPORT_TYPE	ENUMERATED	The transport type supported: 0: L2 1: L3 or higher layer protocols

Annex D MIH protocol message code assignments

(normative)

Table D-1 provides the action identifier (AID) assignment for MIH messages.

Table D-1—AID assignment

MIH messages	AID
MIH messages for Service Management	
MIH_Capability_Discover	1
MIH_Register	2
MIH_DeRegister	3
MIH_Event_Subscribe	4
MIH_Event_Unsubscribe	5
MIH messages for Event Service	
MIH_Link_Detected	1
MIH_Link_Up	2
MIH_Link_Down	3
MIH_Link_Parameters_Report	5
MIH_Link_Going_Down	6
MIH_Link_Handover_Imminent	7
MIH_Link_Handover_Complete	8
MIH messages for Command Service	
MIH_Link_Get_Parameters	1
MIH_Link_Configure_Thresholds	2
MIH_Link_Actions	3
MIH_Net_HO_Candidate_Query	4
MIH_MN_HO_Candidate_Query	5
MIH_N2N_HO_Query_Resources	6
MIH_MN_HO_Commit	7
MIH_Net_HO_Commit	8
MIH_N2N_HO_Commit	9
MIH_MN_HO_Complete	10
MIH_N2N_HO_Complete	11
MIH messages for Information Service	
MIH_Get_Information	1

Table D-2 provides the TLV type value assignment for MIH messages. The type value can be extracted from the binary encoding method of the corresponding data type. TLV type value 110-127 is reserved for experimental TLVs.

Table D-2—Type values for TLV encoding

TLV type name	TLV type value	Data type
Source MIHF ID	1	MIHF_ID
Destination MIHF ID	2	MIHF_ID
Status	3	STATUS
Link type	4	LINK_TYPE
MIH event list	5	MIH_EVT_LIST
MIH command list	6	MIH_CMD_LIST
MIIS query type list	7	MIH_IQ_TYPE_LST
Transport option list	8	MIH_TRANS_LST
Link address list	9	LIST(NET_TYPE_ADDR)
MBB handover support	10	LIST(MBB_HO_SUPP)
Register request code	11	REG_REQUEST_CODE
Valid time interval	12	UNSIGNED_INT(4)
Link identifier	13	LINK_TUPLE_ID
New link identifier	14	LINK_TUPLE_ID
Old access router	15	LINK_ADDR
New access router	16	LINK_ADDR
IP renewal flag	17	IP_RENEWAL_FLAG
Mobility management support	18	IP_MOB_MGMT
IP address configuration methods	19	IP_CFG_MTHDS
Link down reason code	20	LINK_DN_REASON
Time interval	21	UNSIGNED_INT(2)
Link going down reason	22	LINK_GD_REASON
Link parameter report list	23	LIST(LINK_PARAM_RPT)
Device states request	24	DEV_STATES_REQ
Link identifier list	25	LIST(LINK_ID)
Device states response list	26	LIST(DEV_STATES_RSP)
Get status request set	27	LINK_STATUS_REQ
Get status response list	28	LIST(SEQUENCE(LINK_ID,LINK_STATUS_RSP))
Configure request list	29	LIST(LINK_CFG_PARAM)

Table D-2—Type values for TLV encoding

TLV type name	TLV type value	Data type
Configure response list	30	LIST(LINK_CFG_STATUS)
List of link PoA list	31	LIST(LINK_POA_LIST)
Preferred link list	32	LIST(RQ_RESULT)
Handover resource query list	33	QOS_LIST
Handover status	34	HO_STATUS
Access router address	35	IP_ADDR
DHCP server address	36	IP_ADDR
FA address	37	IP_ADDR
IP address information status	38	IP_CFG_STATUS
Link actions list	39	LIST(LINK_ACTION_REQ)
Link actions result list	40	LIST(LINK_ACTION_RSP)
Handover result	41	HO_RESULT
Resource status	42	LINK_RES_STATUS
Resource retention status	43	LINK_RR_STATUS
Info query binary data list	44	LIST(IQ_BIN_DATA)
Info query RDF data list	45	LIST(IQ_RDF_DATA)
Info query RDF schema URL	46	BOOLEAN
Info query RDF schema list	47	LIST(IQ_RDF_SCHM)
Max response size	48	UNSIGNED_INT(2)
Info response binary data list	49	LIST(IR_BIN_DATA)
Info response RDF data list	50	LIST(IR_RDF_DATA)
Info response RDF schema URL list	51	LIST(IR_SCHM_URL)
Info response RDF schema list	52	LIST(IR_RDF_SCHM)
Mobile node MIHF ID	53	MIHF_ID
Query resource report flag	54	BOOLEAN
Event configuration info list	55	LIST(EVT_CFG_INFO)
Target network info	56	TGT_NET_INFO
List of target network info	57	LIST(TGT_NET_INFO)
Assigned resource set	58	ASGN_RES_SET
Link detected info list	59	LIST(LINK_DET_INFO)
MN link ID	60	LINK_ID
PoA	61	LINK_ADDR
Unauthenticated information request	62	BOOLEAN

Table D-2—Type values for TLV encoding

TLV type name	TLV type value	Data type
Network type	63	NETWORK_TYPE
Requested resource set	64	REQ_RES_SET
(Reserved)	65-109	(Reserved)
(Reserved for experimental TLVs)	110-127	(Used for experimental purposes)

Annex E MIIS basic schema

(normative)

The following text defines the RDF vocabularies for MIIS.

```xml
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE rdf:RDF [
 <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
 <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
 <!ENTITY mihbasic "URL_TO_BE_ASSIGNED">
 <!ENTITY owl "http://www.w3.org/2002/07/owl#">
 <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>

<rdf:RDF xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;"
 xmlns:mihbasic="&mihbasic;" xml:base="&mihbasic;"
 xmlns:owl="&owl;" xmlns:xsd="&xsd;">

 <owl:Ontology rdf:about="">
  <rdfs:label>
   Basic Schema for IEEE 802.21 Information Service
  </rdfs:label>
  <owl:versionInfo>1.0</owl:versionInfo>
 </owl:Ontology>

 <owl:DatatypeProperty rdf:ID="ie_type_identifier">
  <rdfs:subPropertyOf rdf:resource="&rdfs;label"/>
  <rdfs:range rdf:resource="&xsd;hexBinary"/>
  <rdfs:comment>
   A type identifier values for Information Elements.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="bit_number">
  <rdfs:range rdf:resource="&xsd;unsignedByte"/>
  <rdfs:comment>
   This property represents a bit number that has
   the value as true.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:ObjectProperty rdf:ID="ie_container_list_of_networks">
  <mihbasic:ie_type_identifier>0x10000300</mihbasic:ie_type_identifier>
  <rdfs:range rdf:resource="#LIST_OF_NETWORKS"/>
 </owl:ObjectProperty>

 <owl:Class rdf:ID="LIST_OF_NETWORKS">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#ie_container_network"/>
    <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:minCardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
 </owl:Class>
```

```
<owl:ObjectProperty rdf:ID="ie_container_network">
 <mihbasic:ie_type_identifier>0x10000301</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#LIST_OF_NETWORKS"/>
 <rdfs:range rdf:resource="#NETWORK"/>
 <rdfs:comment>
  This class contains General Information depicting and Access
  Network Specific Information.
 </rdfs:comment>
</owl:ObjectProperty>

<owl:Class rdf:ID="NETWORK">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#ie_network_type"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#ie_operator_id"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>

<owl:ObjectProperty rdf:ID="ie_network_type">
 <mihbasic:ie_type_identifier>0x10000000</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:range rdf:resource="#NETWORK_TYPE"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="NETWORK_TYPE">
</owl:Class>

<owl:DatatypeProperty rdf:ID="link_type">
 <rdfs:domain rdf:resource="#NETWORK_TYPE"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
 <rdfs:comment>
  Link type of a network. The following values are assigned:
  1: Wireless - GSM
  2: Wireless - GPRS
  3: Wireless - EDGE
  15: Ethernet
  18: Wireless - Other
  19: Wireless - IEEE 802.11
  22: Wireless - CDMA2000
  23: Wireless - UMTS
  24: Wireless - cdma-2000-HRPD
  27: Wireless - IEEE 802.16
  28: Wireless - IEEE 802.20
  29: Wireless - IEEE 802.22
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="revision">
 <rdfs:subPropertyOf rdf:resource="#bit_number"/>
 <rdfs:domain rdf:resource="#NETWORK_TYPE"/>
```

```
  <rdfs:comment>
   The range of #bit_number is 0-63.
  </rdfs:comment>
 </owl:ObjectProperty>

 <owl:DatatypeProperty rdf:ID="type_ext">
  <rdfs:domain rdf:resource="#NETWORK_TYPE"/>
  <rdfs:range rdf:resource="&xsd;string"/>
 </owl:DatatypeProperty>

 <owl:ObjectProperty rdf:ID="ie_operator_id">
  <mihbasic:ie_type_identifier>0x10000001</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="#OPERATOR_ID"/>
 </owl:ObjectProperty>

 <owl:Class rdf:ID="OPERATOR_ID">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#op_name"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#op_namespace"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
 </owl:Class>

 <owl:DatatypeProperty rdf:ID="op_name">
  <rdfs:domain rdf:resource="#OPERATOR_ID"/>
  <rdfs:range rdf:resource="&xsd;string"/>
  <rdfs:comment>
   The value is a non NULL terminated
   string whose length shall not exceed 253 octets.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="op_namespace">
  <rdfs:domain rdf:resource="#OPERATOR_ID"/>
  <rdfs:range rdf:resource="&xsd;unsignedByte"/>
  <rdfs:comment>
   A value of Operator Type:
   0: GSM/UMTS
   1: CDMA
   2: REALM
   3: ITU-T/TSB
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="ie_service_provider_id">
  <mihbasic:ie_type_identifier>0x10000002</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="&xsd;string"/>
  <rdfs:comment>
```

```
   A non-NULL terminated string whose length shall not exceed 253 octets.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="ie_country_code">
  <mihbasic:ie_type_identifier>0x10000003</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="&xsd;string"/>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="ie_network_id">
  <mihbasic:ie_type_identifier>0x10000100</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="&xsd;string"/>
  <rdfs:comment>
   A non-NULL terminated string whose length shall not exceed 253 octets.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="ie_network_aux_id">
  <mihbasic:ie_type_identifier>0x10000101</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="&xsd;string"/>
  <rdfs:comment>
   It is HESSID if network type is IEEE 802.11.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:ObjectProperty rdf:ID="ie_roaming_partners">
  <mihbasic:ie_type_identifier>0x10000102</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="#OPERATOR_ID"/>
 </owl:ObjectProperty>

 <owl:ObjectProperty rdf:ID="ie_cost">
  <mihbasic:ie_type_identifier>0x10000103</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="#COST"/>
 </owl:ObjectProperty>

 <owl:Class rdf:ID="COST">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#cost_unit"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#cost_value"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#cost_curr"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
```

```
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
 </owl:Class>

 <owl:DatatypeProperty rdf:ID="cost_unit">
  <rdfs:domain rdf:resource="#COST"/>
  <rdfs:range rdf:resource="&xsd;unsignedByte"/>
  <rdfs:comment>
   The unit of the cost:
   0: second
   1: minute
   2: hours
   3: day
   4: week
   5: month
   6: year
   7: free
   8: flat rate
   9-255: Reserved
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="cost_value">
  <rdfs:domain rdf:resource="#COST"/>
  <rdfs:range rdf:resource="&xsd;double"/>
  <rdfs:comment>
   The cost value in Currency/Unit
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="cost_curr">
  <rdfs:domain rdf:resource="#COST"/>
  <rdfs:range rdf:resource="&xsd;string"/>
  <rdfs:comment>
   A three-letter currency code(e.g. "USD") specified by
   ISO 4217.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:ObjectProperty rdf:ID="ie_network_qos">
  <mihbasic:ie_type_identifier>0x10000105</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="#QOS_LIST"/>
 </owl:ObjectProperty>

 <owl:Class rdf:ID="QOS_LIST">
 </owl:Class>

 <owl:DatatypeProperty rdf:ID="num_cos_types">
  <rdfs:domain rdf:resource="#QOS_LIST"/>
  <rdfs:range rdf:resource="&xsd;unsignedByte"/>
 </owl:DatatypeProperty>

 <owl:Class rdf:ID="COS">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#cos_id"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
```

```
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#cos_value"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
 </owl:Class>

 <owl:DatatypeProperty rdf:ID="cos_id">
  <rdfs:domain rdf:resource="#COS"/>
  <rdfs:range rdf:resource="&xsd;unsignedByte"/>
  <rdfs:comment>
   A type to represent a class of service identifier.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="cos_value">
  <rdfs:domain rdf:resource="#COS"/>
  <rdfs:range rdf:resource="&xsd;unsignedShort"/>
 </owl:DatatypeProperty>

 <owl:ObjectProperty rdf:ID="min_pk_tx_delay">
  <rdfs:domain rdf:resource="#QOS_LIST"/>
  <rdfs:range rdf:resource="#COS"/>
 </owl:ObjectProperty>

 <owl:ObjectProperty rdf:ID="avg_pk_tx_delay">
  <rdfs:domain rdf:resource="#QOS_LIST"/>
  <rdfs:range rdf:resource="#COS"/>
 </owl:ObjectProperty>

 <owl:ObjectProperty rdf:ID="max_pk_tx_delay">
  <rdfs:domain rdf:resource="#QOS_LIST"/>
  <rdfs:range rdf:resource="#COS"/>
 </owl:ObjectProperty>

 <owl:ObjectProperty rdf:ID="pk_delay_jitter">
  <rdfs:domain rdf:resource="#QOS_LIST"/>
  <rdfs:range rdf:resource="#COS"/>
 </owl:ObjectProperty>

 <owl:ObjectProperty rdf:ID="pk_loss_rate">
  <rdfs:domain rdf:resource="#QOS_LIST"/>
  <rdfs:range rdf:resource="#COS"/>
 </owl:ObjectProperty>

 <owl:DatatypeProperty rdf:ID="ie_network_data_rate">
  <mihbasic:ie_type_identifier>0x10000106</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
  <rdfs:range rdf:resource="&xsd;unsignedInt"/>
 </owl:DatatypeProperty>

 <owl:ObjectProperty rdf:ID="ie_regulat_domain">
  <mihbasic:ie_type_identifier>0x10000107</mihbasic:ie_type_identifier>
  <rdfs:domain rdf:resource="#NETWORK"/>
```

```
 <rdfs:range rdf:resource="#REGULAT_DOMAIN"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="REGULAT_DOMAIN">
</owl:Class>

<owl:DatatypeProperty rdf:ID="regulat_domain_country_code">
 <rdfs:domain rdf:resource="#REGULAT_DOMAIN"/>
 <rdfs:range rdf:resource="&xsd;String"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="regulat_class">
 <rdfs:domain rdf:resource="#REGULAT_DOMAIN"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="ie_net_frequency_bands">
 <mihbasic:ie_type_identifier>0x10000108</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:range rdf:resource="&xsd;unsignedInt"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="ie_net_ip_cfg_methods">
 <mihbasic:ie_type_identifier>0x10000109</mihbasic:ie_type_identifier>
 <rdfs:subPropertyOf rdf:resource="#bit_number"/>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:comment>
  The range of #bit_number is 0-31.
 </rdfs:comment>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="ie_net_capabilities">
 <mihbasic:ie_type_identifier>0x1000010A</mihbasic:ie_type_identifier>
 <rdfs:subPropertyOf rdf:resource="#bit_number"/>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:comment>
  The range of #bit_number is 0-31.
 </rdfs:comment>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="ie_net_supported_lcp">
 <mihbasic:ie_type_identifier>0x1000010B</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="ie_net_mob_mgmt_prot">
 <mihbasic:ie_type_identifier>0x1000010C</mihbasic:ie_type_identifier>
 <rdfs:subPropertyOf rdf:resource="#bit_number"/>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:comment>
  The range of #bit_number is 0-15.
 </rdfs:comment>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="ie_net_emserv_proxy">
 <mihbasic:ie_type_identifier>0x1000010D</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:range rdf:resource="#PROXY_ADDR"/>
```

```
</owl:ObjectProperty>

<owl:Class rdf:ID="PROXY_ADDR">
</owl:Class>

<owl:DatatypeProperty rdf:ID="proxy_addr_ip">
 <rdfs:domain rdf:resource="#PROXY_ADDR"/>
 <rdfs:range rdf:resource="#TRANSPORT_ADDR"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="proxy_addr_fqdn">
 <rdfs:domain rdf:resource="#PROXY_ADDR"/>
 <rdfs:range rdf:resource="&xsd;String"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="ie_net_ims_proxy_cscf">
 <mihbasic:ie_type_identifier>0x1000010E</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:range rdf:resource="#PROXY_ADDR"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="ie_net_mobile_network">
 <mihbasic:ie_type_identifier>0x1000010F</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:range rdf:resource="&xsd;boolean"/>

</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="ie_container_poa">
 <mihbasic:ie_type_identifier>0x10000302</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#NETWORK"/>
 <rdfs:range rdf:resource="#POA"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="POA">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#ie_poa_link_addr"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#ie_poa_location"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#ie_poa_channel_range"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#ie_poa_system_info"/>
```

```
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#ie_poa_subnet_info"/>
   <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:minCardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:comment>
  This class contains all the information depicting a PoA.
 </rdfs:comment>
</owl:Class>

<owl:ObjectProperty rdf:ID="ie_poa_link_addr">
 <mihbasic:ie_type_identifier>0x10000200</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#POA"/>
 <rdfs:range rdf:resource="#LINK_ADDR"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="LINK_ADDR">
</owl:Class>

<owl:Class rdf:ID="LINK_ADDR_MAC_ADDR">
 <rdfs:subClassOf rdf:resource="#LINK_ADDR"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="link_addr_mac_addr">
 <rdfs:subPropertyOf rdf:resource="#addr_value"/>
 <rdfs:domain rdf:resource="#LINK_ADDR_MAC_ADDR"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="LINK_ADDR_3GPP_3G">
 <rdfs:subClassOf rdf:resource="#LINK_ADDR"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="link_addr_3gpp_3g_cell_id_plmn_id">
 <rdfs:domain rdf:resource="#LINK_ADDR_3GPP_3G"/>
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="link_addr_3gpp_3g_cell_id_lac">
 <rdfs:domain rdf:resource="#LINK_ADDR_3GPP_3G"/>
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="link_addr_3gpp_3g_cell_id_ci">
 <rdfs:domain rdf:resource="#LINK_ADDR_3GPP_3G"/>
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="LINK_ADDR_3GPP_2G">
 <rdfs:subClassOf rdf:resource="#LINK_ADDR"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="link_addr_3gpp_2g_cell_id_plmn_id">
 <rdfs:domain rdf:resource="#LINK_ADDR_3GPP_2G"/>
```

```
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="link_addr_3gpp_2g_cell_id_cell_id">
 <rdfs:domain rdf:resource="#LINK_ADDR_3GPP_2G"/>
 <rdfs:range rdf:resource="&xsd;unsignedInt"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="LINK_ADDR_3GPP">
 <rdfs:subClassOf rdf:resource="#LINK_ADDR"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="link_addr_3gpp_addr">
 <rdfs:domain rdf:resource="#LINK_ADDR_3GPP"/>
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="LINK_ADDR_3GPP2">
 <rdfs:subClassOf rdf:resource="#LINK_ADDR"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="link_addr_3gpp2_addr">
 <rdfs:domain rdf:resource="#LINK_ADDR_3GPP2"/>
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="link_addr_other_l2_addr">
 <rdfs:domain rdf:resource="#LINK_ADDR"/>
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="ie_poa_location">
 <mihbasic:ie_type_identifier>0x10000201</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#POA"/>
 <rdfs:range rdf:resource="#LOCATION"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="LOCATION">
</owl:Class>

<owl:Class rdf:ID="BIN_GEO_LOC">
 <rdfs:subClassOf rdf:resource="#LOCATION"/>
 <rdfs:comment>
  This class has properties that represent geographic coordinate.
  The format is based on the Location Configuration Information (LCI)
  defined in IETF RFC 3825.
 </rdfs:comment>
</owl:Class>

<owl:DatatypeProperty rdf:ID="la_res">
 <rdfs:domain rdf:resource="#BIN_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="latitude">
 <rdfs:domain rdf:resource="#BIN_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;double"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="lo_res">
 <rdfs:domain rdf:resource="#BIN_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="longitude">
 <rdfs:domain rdf:resource="#BIN_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;double"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="at">
 <rdfs:domain rdf:resource="#BIN_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
 <rdfs:comment>
  Following codes are defined:
  1: Meters: in 2s-complement fixed-point 22-bit integer part with
  8-bit fraction. If AT = 1, an AltRes value 0.0 would indicate
  unknown altitude. The most precise Altitude would have an AltRes
  value of 30. Many values of AltRes would obscure any variation
  due to vertical datum differences.
  2: Floors: in 2s-complement fixed-point 22-bit integer part with
  8-bit fraction. AT = 2 for Floors enables representing altitude in
  a form more relevant in buildings, which have different
  floor-to-floor dimensions.
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="alt_res">
 <rdfs:domain rdf:resource="#BIN_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
 <rdfs:comment>
  Altitude resolution. six bits indicating the number of valid bits
  in the altitude. Values above 30 (decimal) are undefined and
  reserved.
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="altitude">
 <rdfs:domain rdf:resource="#BIN_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;double"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="datum">
 <rdfs:domain rdf:resource="#BIN_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
 <rdfs:comment>
  Following codes are defined:
  1: WGS
  2: NAD 83 (with associated vertical datum for North American
  vertical datum for 1998)
  3: NAD 83 (with associated vertical datum for Mean Lower Low Water
  (MLLW))
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:Class rdf:ID="XML_GEO_LOC">
 <rdfs:subClassOf rdf:resource="#LOCATION"/>
</owl:Class>
```

```
<owl:DatatypeProperty rdf:ID="xml_geo_loc">
 <rdfs:domain rdf:resource="#XML_GEO_LOC"/>
 <rdfs:range rdf:resource="&xsd;string"/>
 <rdfs:comment>
  Geo address elements as described in IETF RFC 4119.
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:Class rdf:ID="BIN_CIVIC_LOC">
 <rdfs:subClassOf rdf:resource="#LOCATION"/>
 <rdfs:comment>
  This class has properties that represent civic address.
  The format is defined in IETF RFC 4676.
 </rdfs:comment>
</owl:Class>

<owl:DatatypeProperty rdf:ID="civic_cntry_code">
 <rdfs:domain rdf:resource="#BIN_CIVIC_LOC"/>
 <rdfs:range rdf:resource="&xsd;string"/>
 <rdfs:comment>
  Two-letter ISO 3166 country code in capital ASCII letters.
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="civic_addr">
 <rdfs:domain rdf:resource="#BIN_CIVIC_LOC"/>
 <rdfs:range rdf:resource="#CIVIC_ADDR"/>
 <rdfs:comment>
  This property contains the civic address elements.
  The format of the civic address elements is described
  in Section 3.4 of IETF RFC 4676 with a TLV pair
  (whereby the Type and Length fields are one octet long).
 </rdfs:comment>
</owl:ObjectProperty>

<owl:Class rdf:ID="CIVIC_ADDR">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#catype"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#cavalue"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>

<owl:DatatypeProperty rdf:ID="catype">
 <rdfs:domain rdf:resource="#CIVIC_ADDR"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
 <rdfs:comment>
  A one-octet descriptor of the data civic address value.
 </rdfs:comment>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="cavalue">
 <rdfs:domain rdf:resource="#CIVIC_ADDR"/>
 <rdfs:range rdf:resource="&xsd;string"/>
 <rdfs:comment>
  The civic address value.
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:Class rdf:ID="XML_CIVIC_LOC">
 <rdfs:subClassOf rdf:resource="#LOCATION"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="xml_civic_loc">
 <rdfs:domain rdf:resource="#XML_CIVIC_LOC"/>
 <rdfs:range rdf:resource="&xsd;string"/>
 <rdfs:comment>
  Geo address elements as described in IETF RFC 4119.
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:Class rdf:ID="LOCATION_CELL_ID">
 <rdfs:subClassOf rdf:resource="#LOCATION"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="location_cell_id">
 <rdfs:domain rdf:resource="#LOCATION_CELL_ID"/>
 <rdfs:range rdf:resource="&xsd;unsignedInt"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="ie_poa_channel_range">
 <mihbasic:ie_type_identifier>0x10000202</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#POA"/>
 <rdfs:range rdf:resource="#CH_RANGE"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="CH_RANGE">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#low_ch_range"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#high_ch_range"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>

<owl:DatatypeProperty rdf:ID="low_ch_range">
 <rdfs:domain rdf:resource="#CH_RANGE"/>
 <rdfs:range rdf:resource="&xsd;unsignedInt"/>
 <rdfs:comment>
  Lowest channel frequency in MHz
 </rdfs:comment>
```

```
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="high_ch_range">
 <rdfs:domain rdf:resource="#CH_RANGE"/>
 <rdfs:range rdf:resource="&xsd;unsignedInt"/>
 <rdfs:comment>
  Highest channel frequency in MHz
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="ie_poa_system_info">
 <mihbasic:ie_type_identifier>0x10000203</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#POA"/>
 <rdfs:range rdf:resource="#SYSTEM_INFO"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="SYSTEM_INFO">
</owl:Class>

<owl:ObjectProperty rdf:ID="system_info_network_type">
 <rdfs:domain rdf:resource="#SYSTEM_INFO"/>
 <rdfs:range rdf:resource="#NETWORK_TYPE"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="system_info_link_address">
 <rdfs:domain rdf:resource="#SYSTEM_INFO"/>
 <rdfs:range rdf:resource="#LINK_ADDR"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="system_info_parameters">
 <rdfs:domain rdf:resource="#SYSTEM_INFO"/>
 <rdfs:range rdf:resource="#PARAMETERS"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="PARAMETERS">
</owl:Class>

<owl:Class rdf:ID="DCD_UCD">
 <rdfs:subClassOf rdf:resource="#PARAMETERS"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="dcd_ucd_base_id">
 <rdfs:domain rdf:resource="#DCD_UCD"/>
 <rdfs:range rdf:resource="&xsd;unsignedShort"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="bandwidth">
 <rdfs:domain rdf:resource="#DCD_UCD"/>
 <rdfs:range rdf:resource="&xsd;unsignedShort"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="du_ctr_frequency">
 <rdfs:domain rdf:resource="#DCD_UCD"/>
 <rdfs:range rdf:resource="&xsd;unsignedLong"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="eirp">
 <rdfs:domain rdf:resource="#DCD_UCD"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
```

```
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="ttg">
 <rdfs:domain rdf:resource="#DCD_UCD"/>
 <rdfs:range rdf:resource="&xsd;unsignedShort"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="rtg">
 <rdfs:domain rdf:resource="#DCD_UCD"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="down_bp">
 <rdfs:subPropertyOf rdf:resource="#bit_number"/>
 <rdfs:domain rdf:resource="#DCD_UCD"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="up_bp">
 <rdfs:subPropertyOf rdf:resource="#bit_number"/>
 <rdfs:domain rdf:resource="#DCD_UCD"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="init_code">
 <rdfs:domain rdf:resource="#DCD_UCD"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="ho_code">
 <rdfs:domain rdf:resource="#DCD_UCD"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="SIB">
 <rdfs:subClassOf rdf:resource="#PARAMETERS"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="sib_cell_id">
 <rdfs:domain rdf:resource="#SIB"/>
 <rdfs:range rdf:resource="&xsd;unsignedInt"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="fq_code_num">
 <rdfs:domain rdf:resource="#SIB"/>
 <rdfs:range rdf:resource="&xsd;unsignedShort"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="SYS_PARAMS">
 <rdfs:subClassOf rdf:resource="#PARAMETERS"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="sys_params_base_id">
 <rdfs:domain rdf:resource="#SYS_PARAMS"/>
 <rdfs:range rdf:resource="&xsd;unsignedShort"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="pilot_pn">
 <rdfs:domain rdf:resource="#SYS_PARAMS"/>
 <rdfs:range rdf:resource="&xsd;unsignedShort"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="freq_id">
 <rdfs:domain rdf:resource="#SYS_PARAMS"/>
 <rdfs:range rdf:resource="&xsd;unsignedShort"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="band_class">
 <rdfs:domain rdf:resource="#SYS_PARAMS"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="ie_poa_subnet_info">
 <mihbasic:ie_type_identifier>0x10000204</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#POA"/>
 <rdfs:range rdf:resource="#IP_SUBNET_INFO"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="IP_SUBNET_INFO">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#ip_prefix_len"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:resource="#subnet_address"/>
   <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
   </owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>

<owl:DatatypeProperty rdf:ID="ip_prefix_len">
 <rdfs:domain rdf:resource="#IP_SUBNET_INFO"/>
 <rdfs:range rdf:resource="&xsd;unsignedByte"/>
 <rdfs:comment>
  The bit length of the prefix of the subnet to which subnet_address
  property belongs. The prefix_length is less than or equal to 32
  for IPv4 subnet and less than or equal to 128 for IPv6 subnet.
 </rdfs:comment>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="subnet_address">
 <rdfs:domain rdf:resource="#IP_SUBNET_INFO"/>
 <rdfs:range rdf:resource="#TRANSPORT_ADDR"/>
 <rdfs:comment>
  An IP address of the PoA encoded as Address base type defined in
  IETF RFC 3588. The first 2-octet contains AddressType, which may be
  either 1 (IPv4) or 2 (IPv6). If AddressType==1, the subnet_address
  property contains a 4-octet IPv4 address. If AddressType==2, the
  subnet_address property contains a 16-octet IPv6 address.
 </rdfs:comment>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="ie_poa_ip_addr">
 <mihbasic:ie_type_identifier>0x10000205</mihbasic:ie_type_identifier>
 <rdfs:domain rdf:resource="#POA"/>
```

```
  <rdfs:range rdf:resource="#TRANSPORT_ADDR"/>
 </owl:ObjectProperty>

 <owl:Class rdf:ID="TRANSPORT_ADDR">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#addr_family"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#addr_value"/>
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
    </owl:cardinality>
   </owl:Restriction>
  </rdfs:subClassOf>
 </owl:Class>

 <owl:DatatypeProperty rdf:ID="addr_family">
  <rdfs:domain rdf:resource="#TRANSPORT_ADDR"/>
  <rdfs:range rdf:resource="&xsd;unsignedShort"/>
  <rdfs:comment>
   An Address Family defined in
   http://www.iana.org/assignments/address-family-numbers.
  </rdfs:comment>
 </owl:DatatypeProperty>

 <owl:DatatypeProperty rdf:ID="addr_value">
  <rdfs:domain rdf:resource="#TRANSPORT_ADDR"/>
  <rdfs:range rdf:resource="&xsd;hexBinary"/>
  <rdfs:comment>
   An address value specific to address_type.
  </rdfs:comment>
 </owl:DatatypeProperty>
</rdf:RDF>
```

Annex F IEEE 802.21 MIB

(normative)

F.1 Parameters requiring MIB definition

In this standard, only parameters that govern the operation of 802.21 are defined.

F.1.1 MIH capability parameters

Following is a list of parameters that are used in the MIH capability discovery:

— **Link Address List (Read-only)**: A list of network type and link address pairs that are controlled by this MIHF. Note that not all interfaces of an MIH-capable node can be under control of MIHF.

— **MIH Event List (Read-only)**: A list of supported events by this MIHF.

— **MIH Command List (Read-only)**: A list of supported commands by this MIHF.

— **MIH IS Query List (Read-only)**: A list of supported MIH IS query types by this MIHF.

— **MIH Transport List (Read-only)**: A list of supported MIH transport protocols by this MIHF. This is the transport that transmits the MIH protocol messages.

— **List of MBB Handover support (Read-only)**: A list of make-before-break support information for each pair of serving and target network types.

F.1.2 MIH protocol parameters

Following is a list of parameters that are used in the MIH protocol.

— **Local MIHF ID (Read-Write)**

— **List of Peer MIHF IDs (Read-Write)**

— **Transport Type (Read-Write)**

- o L2 or L3
- o Defined for each peer MIHF

— **Version (Read-only)**

— **Maximum Transaction Lifetime (Read-Write)**

- o Unit: seconds

— **Maximum Retransmission Interval (Read-Write)**

- o Unit: seconds

— **Maximum Retransmission Counter (Read-Write)**

- o Unit: none

— **Maximum Average Transmission Rate (Read-Write)**

- o Maximum value of average transmission rate
- o Unit: octets per second

— **Maximum Burst Size (Read-Write)**

- o The maximum number of octets transmitted in a burst
- o Unit: octets

— **aFragmentationThreshold (Read-Write)**

- o Unit: octets

— **ReassemblyTimer (Read-Write)**
  - Unit: seconds

F.2 IEEE 802.21 MIB definition

```
IEEE802dot21-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, OBJECT-TYPE, Integer32, Unsigned32 FROM SNMPv2-SMI
MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF
TEXTUAL-CONVENTION, TruthValue FROM SNMPv2-TC;


-- ***********************************************************************
-- * MODULE IDENTITY
-- ***********************************************************************
ieee802dot21 MODULE-IDENTITY
LAST-UPDATED "200804141225Z"
ORGANIZATION "IEEE 802.21"
CONTACT-INFO
"WG E-mail: stds-802-21@ieee.org
Chair: Vivek G. Gupta
Intel Corporation
E-mail: mailto:vivek.g.gupta@intel.com
Editor: Qiaobing Xie
E-mail: Qiaobing.Xie@MOTOROLA.COM"
DESCRIPTION
"The MIB module for IEEE 802.21 entities.
iso(1).std(0).iso8802(8802).ieee802dot21(21)"
REVISION    "200804141225Z"
DESCRIPTION
"The latest version of this MIB module."
::= { iso std(0) iso8802(8802) ieee802dot21(21) }



-- ***********************************************************************
-- * Textual Conventions
-- ***********************************************************************

Dot21MihfID ::= TEXTUAL-CONVENTION
STATUS      current
DESCRIPTION
"The MIHF ID of an MIH node."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.11"
SYNTAX OCTET STRING (SIZE(0..253))

Dot21LinkType ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d"
STATUS current
DESCRIPTION
"This attribute represents the type of a link."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.4"
SYNTAX Unsigned32 (0..255)
Dot21NetworkRevision ::= TEXTUAL-CONVENTION
```

```
STATUS current
DESCRIPTION
"This attribute represents the network revision of a link."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.8"
SYNTAX OCTET STRING (SIZE(0..8))

Dot21NetworkTypeExtension ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
"This attribute represents a network type extension."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.8"
SYNTAX OCTET STRING (SIZE(0..253))

Dot21EventList ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
"This attribute represents a list of supported events."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.12"
SYNTAX BITS
             { mihLinkDetected(0),
               mihLinkUp(1),
               mihLinkDown(2),
               mihLinkParametersReport(3),
               mihLinkGoingDown(4),
               mihLinkHandoverImminent(5),
               mihLinkHandoverComplete(6),
               mihLinkPDUTransmitStatus(7) }
Dot21CommandList ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
"This attribute represents a list of supported commands."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.12"
SYNTAX BITS
             { mihGetLinkParameters(0),
               mihLinkConfigureThresholds(1),
               mihLinkActions(2),
               mihNetworkHandoverCommands(3),
               mihMobileHandoverCommands(4) }

Dot21ISQueryTypeList ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
" This attribute will be a set of supported MIH IS query types."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.12"
SYNTAX BITS
             { binary(0),
               rdfData(1),
               rdfSchemaUrl(2),
               rdfSchema(3),
               typeIeNetworkType(4),
               typeIeOperatorIdentifier(5),
               typeIeServiceProviderIdentifier(6),
               typeIeCountryCode(7),
               typeIeAccessNetworkIdentifier(8),
               typeIeAccessNetworkAuxIdentifier(9),
               typeIeRoamingPartners(10),
               typeIeCost(11),
               typeIeNetworkQos(12),
               typeIeNetworkDataRate(13),
```

```
                typeIeNetRegulatoryDomain(14),
                typeIeNetFrequencyBands(15),
                typeIeNetIpConfigMethods(16),
                typeIeNetCapabilities(17),
                typeIeNetSupportedLcp(18),
                typeIeNetMobilityMgmtProtocol(19),
                typeIeNetEmergencyServiceProxy(20),
                typeIeNetImsProxyCscf(21),
                typeIeNetMobileNetwork(22),
                typeIePoaLinkAddress(23),
                typeIePoaLocation(24),
                typeIePoaChannelRange(25),
                typeIePoaSystemInformation(26),
                typeIePoaSubnetInformation(27),
                typeIePoaIpAddress(28) }

Dot21TransportList ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
" This attribute will be a set of supported MIH transports."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.12"
SYNTAX BITS { udp(0), tcp(1) }

-- *********************************************************************
-- * Major sections
-- *********************************************************************
-- MIH Function Management (MIHMT) Attributes
-- DEFINED AS "The MIHMT object class provides the necessary support
-- at the MIHF to manage the processes in the station such that
-- the MIHF can work cooperatively as a part of an IEEE 802.21
-- network."
dot21mihmt OBJECT IDENTIFIER ::= { ieee802dot21 1 }
-- dot21mihmt GROUPS
-- dot21LocalMihfTable ::= { dot21mihmt 1 }
-- dot21PeerMihfTable ::= { dot21mihmt 2 }
-- dot21MbbHandoverSupportTable ::= { dot21mihmt 3 }

-- *********************************************************************
-- * MIB attribute OBJECT-TYPE definitions follow
-- *********************************************************************

--
-- Local MIHF Table
--

dot21LocalMihfTable OBJECT-TYPE
SYNTAX SEQUENCE OF Dot21LocalMihfEntry
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
"The table of local MIHFs. The MIH MIB allows to have more than one local MIHFs
per SNMP engine."
::={ dot21mihmt 1 }

dot21LocalMihfEntryOBJECT-TYPE
SYNTAX Dot21LocalMihfEntry
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
```

```
"The value contains information associated with a particular local MIHF. In most
cases, there will be only one local MIHF on a node."
INDEX  { dot21LocalMihfIndex }
::={ dot21LocalMihfTable 1}

Dot21LocalMihfEntry ::=
SEQUENCE{
dot21LocalMihfIndex Integer32,
dot21LocalMihfID Dot21MihfID,
dot21LocalEventList Dot21EventList,
dot21LocalCommandList Dot21CommandList,
dot21LocalISQueryTypeList Dot21ISQueryTypeList,
dot21LocalTransportList Dot21TransportList,
dot21LocalVersion Integer32,
dot21LocalMaxTransactionLifetime Integer32,
dot21LocalMaxRetransmissionIntvl Integer32,
dot21LocalMaxRetransmissionCntr Integer32,
dot21LocalMaxAvgTransmissionRate Integer32,
dot21LocalMaxBurstSize Integer32,
dot21LocalFragmentationThreshold Integer32,
dot21LocalReassemblyTimeout Integer32
}

dot21LocalMihfIndex OBJECT-TYPE
SYNTAX Integer32(0..2147483647)
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
"Index of local MIHF table."
::= { dot21LocalMihfEntry 1 }

dot21LocalMihfID OBJECT-TYPE
SYNTAX Dot21MihfID
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"The MIHF ID of this node."
REFERENCE "IEEE Std 802.21, 2008 Edition, C.3.11"
::={ dot21LocalMihfEntry 2 }

dot21LocalEventList OBJECT-TYPE
SYNTAX Dot21EventList
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This attribute will be a set of all the MIH events supported by this MIH node."
DEFVAL { {} }
::={ dot21LocalMihfEntry 3 }

dot21LocalCommandList OBJECT-TYPE
SYNTAX Dot21CommandList
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This attribute will be a set of all the MIH commands supported by this MIH
node."
DEFVAL { {} }
::={ dot21LocalMihfEntry 4 }
```

```
dot21LocalISQueryTypeList OBJECT-TYPE
SYNTAX Dot21ISQueryTypeList
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This attribute will be a set of MIH IS query types supported by this MIH node."
DEFVAL { {} }
::={ dot21LocalMihfEntry 5 }

dot21LocalTransportList OBJECT-TYPE
SYNTAX Dot21TransportList
MAX-ACCESSread-only
STATUS current
DESCRIPTION
" This attribute will be a set of MIH transports supported by this MIH node."
DEFVAL { {} }
::={ dot21LocalMihfEntry 6 }


dot21LocalVersion OBJECT-TYPE
SYNTAX Unsigned32 (1..15)
MAX-ACCESSread-only
STATUS current
DESCRIPTION
"The MIH protocol version supported by this MIHF."
DEFVAL { 1 }
::={ dot21LocalMihfEntry 7 }

dot21LocalMaxTransactionLifetime OBJECT-TYPE
SYNTAX Unsigned32 (1..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The maximum time in seconds for a MIH protocol transaction."
DEFVAL { 30 }
::={ dot21LocalMihfEntry 8 }

dot21LocalMaxRetransmissionIntvl OBJECT-TYPE
SYNTAX Unsigned32 (1..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The maximum time in seconds for retransmitting a MIH message."
DEFVAL { 10 }
::={ dot21LocalMihfEntry 9 }

dot21LocalMaxRetransmissionCntr OBJECT-TYPE
SYNTAX Unsigned32 (1..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The maximum number of retransmission retries for MIH messages."
DEFVAL { 2 }
::={ dot21LocalMihfEntry 10 }

dot21LocalMaxAvgTransmissionRate OBJECT-TYPE
SYNTAX Unsigned32 (0..255)
MAX-ACCESSread-write
STATUS current
```

```
DESCRIPTION
"The maximum number of MIH messages can be transmitted per second on this node.
If the value is 0, no limitation is set."
DEFVAL { 0 }
::={ dot21LocalMihfEntry 11 }

dot21LocalMaxBurstSize OBJECT-TYPE
SYNTAX Unsigned32 (0..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
" The maximum number of octets transmitted in a burst. If the value is 0, no lim-
itation is set."
DEFVAL { 0 }
::={ dot21LocalMihfEntry 12 }

dot21LocalFragmentationThreshold OBJECT-TYPE
SYNTAX Unsigned32 (8..65535)
MAX-ACCESS read-write
STATUS current
DESCRIPTION "The value for aFragmentationThreshold on this node."
DEFVAL { 1500 }
::={ dot21LocalMihfEntry 13 }

dot21LocalReassemblyTimeout OBJECT-TYPE
SYNTAX Unsigned32 (1..255)
MAX-ACCESS read-write
STATUS current
DESCRIPTION "The timeout value for ReassemblyTimer."
DEFVAL { 5 }
::={ dot21LocalMihfEntry 14 }


--
-- The Peer MIHF Table
--

dot21PeerMihfTable OBJECT-TYPE
SYNTAX SEQUENCE OF Dot21PeerMihfEntry
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
"The table of MIHF known by this MIHF."
::={ dot21mihmt 2 }

dot21PeerMihfEntryOBJECT-TYPE
SYNTAX Dot21PeerMihfEntry
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
"Details of a specific MIHF peer."
INDEX  {dot21PeerMihfIndex}
::= { dot21PeerMihfTable 1 }

Dot21PeerMihfEntry ::=
SEQUENCE {
dot21PeerMihfIndex Integer32,
dot21PeerMihfIDDot21MihfID,
dot21PeerLocalMihfID Dot21MihfID,
```

```
dot21PeerEventList Dot21EventList,
dot21PeerCommandList Dot21CommandList,
dot21PeerISQueryTypeList Dot21ISQueryTypeList,
dot21PeerTransportList Dot21TransportList,
dot21PeerTransportTypeINTEGER,
dot21PeerVersion Integer32,
dot21PeerMaxTransactionLifetime Integer32,
dot21PeerMaxRetransmissionIntvl Integer32,
dot21PeerMaxRetransmissionCntr Integer32,
dot21PeerMaxAvgTransmissionRate Integer32,
dot21PeerMaxBurstSize Integer32,
dot21PeerFragmentationThreshold Integer32,
dot21PeerReassemblyTimeout Integer32
}

dot21PeerMihfIndex OBJECT-TYPE
SYNTAX Integer32(0..2147483647)
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
"Index of peer MIHF table."
::= { dot21PeerMihfEntry 1 }


dot21PeerMihfID OBJECT-TYPE
SYNTAX Dot21MihfID
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The MIHF ID of a peer MIH node."
::={ dot21PeerMihfEntry 2 }

dot21PeerLocalMihfID OBJECT-TYPE
SYNTAX Dot21MihfID
MAX-ACCESSread-only
STATUS current
DESCRIPTION
"The MIHF ID of the local MIH node for this peer MIH node."
::={ dot21PeerMihfEntry 3 }


dot21PeerEventList OBJECT-TYPE
SYNTAX Dot21EventList
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This attribute will be a set of all the MIH events supported by peer MIH node."
DEFVAL { {} }
::={ dot21PeerMihfEntry 4 }

dot21PeerCommandList OBJECT-TYPE
SYNTAX Dot21CommandList
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This attribute will be a set of all the MIH commands supported by peer MIH
node."
DEFVAL { {} }
::={ dot21PeerMihfEntry 5 }
```

```
dot21PeerISQueryTypeList OBJECT-TYPE
SYNTAX Dot21ISQueryTypeList
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This attribute will be a set of MIH IS query types supported by peer MIH node."
DEFVAL { {} }
::={ dot21PeerMihfEntry 6 }

dot21PeerTransportList OBJECT-TYPE
SYNTAX Dot21TransportList
MAX-ACCESSread-only
STATUS current
DESCRIPTION
" This attribute will be a set of MIH transports supported by peer MIH  node."
DEFVAL { {} }
::={ dot21PeerMihfEntry 7 }

dot21PeerTransportType OBJECT-TYPE
SYNTAX INTEGER { layerTwo(2), layerThree(3) }
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"This value should be set for the MIH protocol layer used for transmitting MIH
messages."
DEFVAL { layerTwo }
::= {dot21PeerMihfEntry 8 }

dot21PeerVersion OBJECT-TYPE
SYNTAX Unsigned32 (1..15)
MAX-ACCESSread-only
STATUS current
DESCRIPTION
"The MIH protocol version supported by peer MIHF. The default version is 1."
       DEFVAL { 1 }
::={ dot21PeerMihfEntry 9 }

dot21PeerMaxTransactionLifetime OBJECT-TYPE
SYNTAX Unsigned32 (1..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The maximum time in seconds for a MIH protocol transaction used for a  particu-
lar peer MIHF."
DEFVAL { 30 }
::={ dot21PeerMihfEntry 10 }

dot21PeerMaxRetransmissionIntvl OBJECT-TYPE
SYNTAX Unsigned32 (1..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The maximum time in seconds for retransmitting a MIH message used for a partic-
ular peer MIHF."
DEFVAL { 10 }
::={ dot21PeerMihfEntry 11 }

dot21PeerMaxRetransmissionCntr OBJECT-TYPE
```

```
SYNTAX Unsigned32 (1..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The maximum number of retransmission retries for MIH messages used for a partic-
ular peer MIHF."
DEFVAL { 2 }
::={ dot21PeerMihfEntry 12 }

dot21PeerMaxAvgTransmissionRate OBJECT-TYPE
SYNTAX Unsigned32 (0..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The maximum number of MIH messages can be transmitted per second on this node
for a particular peer MIHF. If the value is 0, no limitation is set."
DEFVAL { 0 }
::={ dot21PeerMihfEntry 13 }

dot21PeerMaxBurstSize OBJECT-TYPE
SYNTAX Unsigned32 (0..255)
MAX-ACCESSread-write
STATUS current
DESCRIPTION
"The maximum number of octets transmitted in a burst. If the value is 0, no lim-
itation is set."
DEFVAL { 0 }
::={ dot21PeerMihfEntry 14 }

dot21PeerFragmentationThreshold OBJECT-TYPE
SYNTAX Unsigned32 (8..65535)
MAX-ACCESS read-write
STATUS current
DESCRIPTION "The value for aFragmentationThreshold used for this peer MIH node."
DEFVAL { 1500 }
::={ dot21LocalMihfEntry 15 }

dot21PeerReassemblyTimeout OBJECT-TYPE
SYNTAX Unsigned32 (1..255)
MAX-ACCESS read-write
STATUS current
DESCRIPTION "The timeout value for ReassemblyTimer used for this peer MIH node."
DEFVAL { 5 }
::={ dot21LocalMihfEntry 16 }


--
-- The Make-Before-Break Handover Support Table
--

dot21MbbHandoverSupportTable OBJECT-TYPE
SYNTAX SEQUENCE OF Dot21MbbHandoverSupportEntry
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
"The table of make-before-break handover support entries."
::={ dot21mihmt 4 }

dot21MbbHandoverSupportEntryOBJECT-TYPE
```

```
SYNTAX Dot21MbbHandoverSupportEntry
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
"The value contains information associated with a particular MBB support."
INDEX  { dot21MbbHandoverSupportIndex }
::={ dot21MbbHandoverSupportTable 1 }

Dot21MbbHandoverSupportEntry ::=
SEQUENCE{
dot21MbbHandoverSupportIndex Integer32,
dot21FromLinkType Dot21LinkType,
dot21FromNetworkRevision Dot21NetworkRevision,
dot21FromNetworkTypeExtension Dot21NetworkTypeExtension,
dot21ToLinkType Dot21LinkType,
dot21ToNetworkRevision Dot21NetworkRevision,
dot21ToNetworkTypeExtension Dot21NetworkTypeExtension,
dot21IsMbbSupported TruthValue
}

dot21MbbHandoverSupportIndex OBJECT-TYPE
SYNTAX Integer32(0..2147483647)
MAX-ACCESSnot-accessible
STATUS current
DESCRIPTION
"Index of make-before-break handover support table."
::= { dot21MbbHandoverSupportEntry 1 }

dot21FromLinkType OBJECT-TYPE
SYNTAX Dot21LinkType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This attribute represents the link type of serving link."
DEFVAL { 0 }
::={ dot21MbbHandoverSupportEntry 2 }

dot21FromNetworkRevision OBJECT-TYPE
SYNTAX Dot21NetworkRevision
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This attribute represents the network revision of serving link."
DEFVAL { ''H }
::={ dot21MbbHandoverSupportEntry 3 }

dot21FromNetworkTypeExtension OBJECT-TYPE
SYNTAX Dot21NetworkTypeExtension
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This attribute represents the network type extension of serving link."
DEFVAL { ''H }
::={ dot21MbbHandoverSupportEntry 4 }

dot21ToLinkType OBJECT-TYPE
SYNTAX Dot21LinkType
MAX-ACCESS read-only
STATUS current
```

```
DESCRIPTION
"This attribute represents the link type of target link."
DEFVAL { 0 }
::={ dot21MbbHandoverSupportEntry 5 }

dot21ToNetworkRevision OBJECT-TYPE
SYNTAX Dot21NetworkRevision
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This attribute represents the network revision of target link."
DEFVAL { ''H }
::={ dot21MbbHandoverSupportEntry 6 }

dot21ToNetworkTypeExtension OBJECT-TYPE
SYNTAX Dot21NetworkTypeExtension
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This attribute represents the network type extension of target link."
DEFVAL { ''H }
::={ dot21MbbHandoverSupportEntry 7 }

dot21IsMbbSupported OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This attribute indicates whether make-before-break handover is supported. A
value of true indicates that make-before-break handover is supported. A value of
false indicates that make-before-break handover is not supported."
::={ dot21MbbHandoverSupportEntry 8 }

-- ********************************************************************
-- * Conformance Information
-- ********************************************************************
dot21Conformance OBJECT IDENTIFIER ::= { ieee802dot21 2 }
dot21Groups OBJECT IDENTIFIER ::= { dot21Conformance 1 }
dot21Compliances OBJECT IDENTIFIER ::= { dot21Conformance 2 }

-- ********************************************************************
-- * Compliance Statements
-- ********************************************************************
dot21Compliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"The compliance statement for SNMPv2 entities that implement
the IEEE 802.21 MIB."
MODULE -- this module
MANDATORY-GROUPS {
  dot21MihmtBase1
}
::= { dot21Compliances 1 }
dot21MihmtBase1 OBJECT-GROUP
OBJECTS {
  dot21LocalMihfID,
  dot21LocalEventList,
  dot21LocalCommandList,
  dot21LocalISQueryTypeList,
```

```
  dot21LocalTransportList,
  dot21LocalVersion,
  dot21LocalMaxTransactionLifetime,
  dot21LocalMaxRetransmissionIntvl,
  dot21LocalMaxRetransmissionCntr,
  dot21LocalMaxAvgTransmissionRate,
  dot21LocalMaxBurstSize,
  dot21PeerMihfID,
  dot21PeerLocalMihfID,
  dot21PeerEventList,
  dot21PeerCommandList,
  dot21PeerISQueryTypeList,
  dot21PeerTransportList,
  dot21PeerTransportType,
  dot21PeerVersion,
  dot21PeerMaxTransactionLifetime,
  dot21PeerMaxRetransmissionIntvl,
  dot21PeerMaxRetransmissionCntr,
  dot21PeerMaxAvgTransmissionRate,
  dot21PeerMaxBurstSize,
  dot21FromLinkType,
  dot21FromNetworkRevision,
  dot21FromNetworkTypeExtension,
  dot21ToLinkType,
  dot21ToNetworkRevision,
  dot21ToNetworkTypeExtension,
  dot21IsMbbSupported
}
STATUS current
DESCRIPTION
"This object class provides the necessary support at the MIH node to manage the
processes in the MIH node so that the MIH node may work cooperatively as a part
of an IEEE 802.21 network."
::= { dot21Groups 1 }

END
```

Annex G Protocol implementation conformance statement (PICS) proforma[3]

(normative)

G.1 Introduction

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given IEEE 802 standard. Such a statement is called an Implementation Conformance Statement (ICS).

G.2 Scope

This annex provides the ICS proforma for the IEEE Std 802.21 in compliance with the relevant requirements, and in accordance with the relevant guidance, given in ITU-T Recommendation X.290 (1995), OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications - General concept and.ITU-T Recommendation X.296 (1995) OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications - Implementation conformance statements.

G.3 Conformance

If it is claimed to conform to IEEE Std 802.21, the actual PICS Proforma to be filled in by a supplier shall be technically equivalent to the text of the PICS Proforma in this annex, and shall preserve the number/naming and ordering of the PICS Proforma items.

A PICS that conforms to this IEEE Std 802.21 shall be a conforming PICS proforma completed in accordance with the instructions for completion given in G.4.

G.4 Instructions

G.4.1 Purpose and structure

The supplier of a protocol implementation that is claimed to conform to IEEE Std 802.21, shall complete the following protocol implementation conformance statement (PICS) proforma.

The PICS proforma expresses in compact form the static conformance requirements of this standard. It serves as a reference to the static conformance review. ITU-T Recommendation X.296, 6.7, provides examples of uses and users of proformas.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented.

[3]Copyright release for PICS proformas: Users of this standard may freely reporoduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

This PICS proforma has the following structure. Within this, the instructions subclause, are the purpose and scope; symbols, abbreviations, and terms; and explicit instructions for completing the implementation conformance statement. After the instructions are the subclauses for identifying the implementation, the protocol, and corrigenda (if any). The final subclause contains the questionnaire in tabular form. Within this final subclause a separate table is used to cover these categorizes: global statement of conformance, roles, major capabilities, protocol data units (PDUs), PDU parameters, and timers.

The structure of the individual tables varies. Except for the tables for the identification of the protocol and the identification of any corrigenda, at a minimum all tables have columns for item number, item description, status, and support. For most the status columns includes both a status and a predicate. Some tables have a column for a mnemonic, which makes for easier cross-referencing within the PICS proforma. Most tables contain a reference column. A few tables (e.g., PDU parameters and timers) contain a column for entering supported value(s).

G.4.2 Symbols, abbreviations, and terms

M mandatory
O optional
O.<n> optional, but support of at least one of the group of options labeled by the same numeral <n> is required
pred: conditional symbol, including predicate identification (mnemonic)

N/A not applicable

GBLx mnemonic for global statement of conformance, where x is an integer

MCx mnemonic for major capabilities, where x is an integer

PDUx mnemonic for protocol data unit, where x is an integer

RLx mnemonic for role, where x is an integer

G.4.3 Explicit instructions

The blank spaces in the identification of the implementation part is to be completed with the information necessary to identify fully both the implementation and supplier, as well as the name of a person to contact if there are any queries concerning the contents of the PICS

For the identification of the protocol, indicate in the support column “Yes”, if this is the protocol being supported.

For the identification of corrigenda to the protocol, indicate if any corrigenda have been applied by entering the corrigenda information in the space provided. If none are applicable, then leave it blank.

The main part of the PICS proforma is a fixed questionnaire in tabular form, divided into subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the column labelled support, either by simply marking an answer to indicate a restricted choice (i.e., Yes or No) or by entering a value or a set or a range of values in the supported range column. (Note that there are some items where two or more choices from a set of possible answers may apply. All relevant choices are to be marked in these cases.)

Each item is identified by an item number, which is given in the first column. The second column (labelled item description) contains the question to be answered. The remaining columns may be labeled: references, status, support, allowed value(s), supported value(s), or mnemonic. The reference column contains the reference or references to the appropriate static conformance requirements or other clauses in the IEEE Std 802.21. The status column contains the status value (mandatory, optional, not applicable, or conditional (see F.4.6)) of the item. The answer to the item is to be entered in the support column by either entering Yes or No in the space provided or, if present, mark the appropriate tick box beside the appropriate answer. For items that contain a supported value(s) column, in addition to answering the support column, enter the value or values supported for the item in the space provided.

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

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

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

G.4.4 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist in the interpretation of the PICS. It is not intended or expected that a large amount of information will be supplied, and a PICS can be considered complete without any such information. Examples of such Additional Information might be an outline of the ways in which an (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this standard but have a bearing upon the answers to some items.

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

G.4.5 Exception information

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

An implementation for which an Exception Information item is required in this way does not conform to this standard.

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

G.4.6 Conditional status

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

In this PICS proforma conditional items are represented through the use of nesting item numbering and by using individual conditional items as indicated in the status column.

If the value of the predicate is true, the conditional item is applicable, and its status is given by S and the support column is to be completed in the usual way. Otherwise, the conditional item is not relevant and the N/A answer is to be marked.

A predicate is an mnemonic for an item in the PICS proforma. The value of the predicate is true if the item is marked as supported, and is false otherwise.

Each item referenced in a predicate is indicated by an asterisk in the item number column.

G.5 Identification of the implementation

In the space provided in the following subclauses provide all information that will uniquely identify both the supplier (or client of the test laboratory) and the implementation and the system on which it resides. Also provide a person as a point of contact for any queries concerning the contents of the PICS.

G.5.1 Implementation and the system

G.5.2 Supplier or client of the test laboratory

Name:

Address:

G.5.3 Contact

Name:

Address:

G.6 Identification of the protocol

Item number	Identification of protocol specification	Support
G.6.1	IEEE Std 802.21 (2009)	

G.7 Identification of corrigenda to the protocol

Identification of corrigenda implemented	
Specification	**Corrigenda implemented**
IEEE Std 802.21 (2009)	

G.8 PICS proforma tables

G.8.1 Global statement of conformance

Item number	Item description	Status	Support	Mnemonic
G.8.1	Have all mandatory capabilities been implemented?	M	Yes [ ] No [ ]	GBL1

Answering “No” to this question indicates non-conformance to the IEEE Std 802.21. Non-supported mandatory capabilities are to be identified in the implementation conformance statement, with an explanation of why the implementation is non-conforming.

G.8.2 Roles

Item number	Item description	References	Status	Support	Mnemonic
G.8.2.1	Is MIHF supported in a mobile node?	5.4	O.1	Yes [ ] No [ ]	RL1
G.8.2.2	Is MIHF supported in a network entity?	5.4	O.1	Yes [ ] No [ ]	RL2

G.8.3 Major capabilities

Item number	Item description	References	Status	Support	Mnemonic
*G.8.3.1	Is the Event Service (ES) supported?	6.3	O.2	Yes [ ] No [ ]	MC1
*G.8.3.2	Is the Command Service (CS) supported?	6.4	O.2	Yes [ ] No [ ]	MC2

*G.8.3.3	Is the Information Service (IS) supported?	6.5	O.2	Yes [ ] No [ ]	MC3
G.8.3.3.1	Is the TLV query method supported?	6.5.6.1, 6.5.6.2	MC3:O.3	Yes [ ] No [ ] N/A [ ]	MC3.1
G.8.3.3.2	Is the RDF query method supported?	6.5.6.1, 6.5.6.3	MC3:O.3	Yes [ ] No [ ] N/A [ ]	MC3.2
*G.8.3.4	Is Capability Discovery supported?	6.2.3	M	Yes [ ] No [ ]	MC4
G.8.3.4.1	Is Unsolicited Capability Discovery supported?	8.2.4.3.3	O	Yes [ ] No [ ]	MC4.1
G.8.3.4.2	Is Solicited Capability Discovery supported?	8.2.4.3.4	M	Yes [ ] No [ ]	MC4.2
*G.8.3.5	Is the Registration Service supported?	6.2.4	O	Yes [ ] No [ ]	MC5
G.8.3.6	Is Mobile Initiated Handover supported?	6.4.3.2.3	O.4	Yes [ ] No [ ]	MC6
G.8.3.7	Is Network Initiated Handover supported?	6.4.3.2.4	O.4	Yes [ ] No [ ]	MC7
G.8.3.8	Is MIH Acknowledgement protocol supported?	8.2.2	O	Yes [ ] No [ ]	MC8
G.8.3.9	Is MIH fragmentation supported?	8.4.2	M	Yes [ ] No [ ]	MC9

G.8.4 PDUs

Item number	Item description	References	Status	Support	Mnemonic
G.8.4.1	MIH_Link_Detected indication?	8.6.2.1	MC1:M	Yes [ ] No [ ] N/A[ ]	PDU1
G.8.4.2	MIH_Link_Up indication?	8.6.2.2	MC1:M	Yes [ ] No [ ] N/A[ ]	PDU2
G.8.4.3	MIH_Link_Down indication?	8.6.2.3	MC1:M	Yes [ ] No [ ] N/A[ ]	PDU3
G.8.4.4	MIH_Link_Going_Down indication?	8.6.2.5	MC1:M	Yes [ ] No [ ] N/A[ ]	PDU4
G.8.4.5	MIH_Link_Parameters_Report indication?	8.6.2.4	MC1:M	Yes [ ] No [ ] N/A[ ]	PDU5
G.8.4.6	MIH_Link_Handover_Imminent indication?	8.6.2.6	MC1:O	Yes [ ] No [ ] N/A[ ]	PDU6
G.8.4.7	MIH_Link_Handover_Complete indication?	8.6.2.7	MC1:O	Yes [ ] No [ ] N/A[ ]	PDU7
G.8.4.8	MIH_Link_Get_Parameters request?	5.3.3.1, 8.6.3.1	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU8
G.8.4.9	MIH_Link_Get_Parameters response?	5.3.3.1, 8.6.3.2	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU9

G.8.4.10	MIH_Link_Configure_Thresholds request?	5.3.3.1, 8.6.3.3	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU10
G.8.4.11	MIH_Link_Configure_Thresholds response?	5.3.3.1, 8.6.3.4	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU11
G.8.4.12	MIH_Link_Action request?	5.3.3.1, 8.6.3.5	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU12
G.8.4.13	MIH_Link_Action response?	5.3.3.1, 8.6.3.6	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU13
G.8.4.14	MIH_Net_HO_Candidate_Query request?	5.3.3.1, 8.6.3.7	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU14
G.8.4.15	MIH_Net_HO_Candidate_Query response?	5.3.3.1, 8.6.3.8	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU15
G.8.4.16	MIH_MN_HO_Candidate_Query request?	5.3.3.1, 8.6.3.9	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU16
G.8.4.17	MIH_MN_HO_Candidate_Query response?	5.3.3.1, 8.6.3.10	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU17
G.8.4.18	MIH_N2N_HO_Query_Resources request?	5.3.3.1, 8.6.3.11	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU18
G.8.4.19	MIH_N2N_HO_Query_Resources response?	5.3.3.1, 8.6.3.12	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU19
G.8.4.20	MIH_MN_HO_Commit request?	5.3.3.1, 8.6.3.13	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU20
G.8.4.21	MIH_MN_HO_Commit response?	5.3.3.1, 8.6.3.14	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU21
G.8.4.22	MIH_Net_HO_Commit request?	5.3.3.1, 8.6.3.15	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU22
G.8.4.23	MIH_Net_HO_Commit response?	5.3.3.1, 8.6.3.16	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU23
G.8.4.24	MIH_N2N_HO_Commit request?	5.3.3.1, 8.6.3.17	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU24
G.8.4.25	MIH_N2N_HO_Commit response?	5.3.3.1, 8.6.3.18	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU25
G.8.4.26	MIH_MN_HO_Complete request?	5.3.3.1, 8.6.3.19	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU26
G.8.4.27	MIH_MN_HO_Complete response?	5.3.3.1, 8.6.3.20	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU27
G.8.4.28	MIH_N2N_HO_Complete request?	5.3.3.1, 8.6.3.21	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU28
G.8.4.29	MIH_N2N_HO_Complete response?	5.3.3.1, 8.6.3.22	MC2:M	Yes [ ] No [ ] N/A[ ]	PDU29
G.8.4.30	MIH_Get_Information request?	8.6.4.1	MC3:M	Yes [ ] No [ ] N/A[ ]	PDU30
*G.8.4.31	MIH_Get_Information response?	8.6.4.2	MC3:M	Yes [ ] No [ ] N/A[ ]	PDU31
G.8.4.32	MIH_Capability_Discover request?	8.6.1.1	MC4:M	Yes [ ] No [ ] N/A[ ]	PDU32

G.8.4.33	MIH_Capability_Discover response?	8.6.1.2	MC4:M	Yes [ ] No [ ] N/A[ ]	PDU33
G.8.4.34	MIH_Register request?	8.6.1.3	MC5:M	Yes [ ] No [ ] N/A[ ]	PDU34
G.8.4.35	MIH_Register response?	8.6.1.4	MC5:M	Yes [ ] No [ ] N/A[ ]	PDU35
G.8.4.36	MIH_DeRegister request?	8.6.1.5	MC5:M	Yes [ ] No [ ] N/A[ ]	PDU36
G.8.4.37	MIH_DeRegister response?	8.6.1.6	MC5:M	Yes [ ] No [ ] N/A[ ]	PDU37
G.8.4.38	MIH_Event_Subscribe request?	8.6.1.7	M	Yes [ ] No [ ]	PDU38
G.8.4.39	MIH_Event_Subscribe response?	8.6.1.8	M	Yes [ ] No [ ]	PDU39
G.8.4.40	MIH_Event_Unsubscribe request?	8.6.1.9	M	Yes [ ] No [ ]	PDU40
G.8.4.41	MIH_Event_Unsubscribe response?	8.6.1.10	M	Yes [ ] No [ ]	PDU41

G.8.5 PDU parameters

Item number	Item description	References	Status	Support	Supported Value
G.8.5.1	Maximum supported MIH_Get_Information response message size (in octets)?	8.6.4.2	PDU31: M	Yes [ ] No [ ] N/A [ ]	

G.8.6 Timers

Item number	Item description	References	Status	Support	Allowed values (s)	Supported values (s)
G.8.6.1	TransactionLifeTime Timer	8.2.3.5	M	Yes [ ] No [ ]		
G.8.6.2	RetransmissionInterval Timer	8.2.3.8.2	M	Yes [ ] No [ ]		
G.8.6.3	ReassemblyTimer	8.4.2.3	M	Yes [ ] No [ ]		

Annex H Mapping MIH messages to reference points

(normative)

Table H-1—Mapping MIH messages to reference points

MIH message name	Reference point
MIH_Capability_Discover	RP1, RP2, RP3, RP4, RP5
MIH_Event_Subscribe	RP1, RP3
MIH_Event_Unsubscribe	RP1, RP3
MIH_Register	RP1, RP3, RP5
MIH_DeRegister	RP1, RP3, RP5
MIH_Link_Detected	RP1, RP3
MIH_Link_Up	RP1, RP3, RP5
MIH_Link_Down	RP1, RP3, RP5
MIH_Link_Parameters_Report	RP1, RP3, RP5
MIH_Link_Going_Down	RP1, RP3, RP2
MIH_Link_Handover_Imminent	RP1, RP3, RP2
MIH_Link_Handover_Complete	RP1, RP3
MIH_Link_Get_Parameters	RP1, RP3, RP2
MIH_Link_Configure_Thresholds	RP1, RP3
MIH_Link_Actions	RP1, RP3
MIH_Net_HO_Candidate_Query	RP1, RP3
MIH_MN_HO_Candidate_Query	RP1, RP3
MIH_N2N_HO_Query_Resources	RP5
MIH_MN_HO_Commit	RP1, RP3
MIH_Net_HO_Commit	RP1, RP3
MIH_N2N_HO_Commit	RP5
MIH_MN_HO_Complete	RP1, RP2, RP3

Table H-1—Mapping MIH messages to reference points

MIH message name	Reference point
MIH_N2N_HO_Complete	RP5
MIH_Get_Information	RP1, RP2, RP3, RP4, RP5

Annex I Quality of service mapping

(normative)

This annex provides the mapping between 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 I-1, Table I-2, and Table I-3 show the mapping between generic QoS parameters and those used by different technologies such as 802.11, 802.16 and 3GPP. I.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 termed links in this standard.

The maximum attainable speed of information transfer over a given communication channel can be constant, as it is usually the case with network segments involving only wired links, or it can be time varying at different scales, as is often the case for segments involving wireless links. This measure will be 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: (*) If the link supports re-transmission, then the Packet Error Rate includes it, otherwise it does not include it.

For a link that supports CoS differentiation, per 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: representing 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.

I.1 Generic IEEE 802.21 QoS flow diagram

Figure I-1 represents an example flow diagram for using the QoS framework defined by the MIHF.

The following terms are used in Figure I-1:

- UP Entity: An upper layer entity such as a multimedia application;
- MAC-S: The MAC layer of the interface that is currently serving the MN;
- MAC-C: The MAC layer of an interface that is not currently serving the MN;
- PoA-S: The serving PoA;
- PoS-S: The serving PoS.

The MIH_Link_Configure_Thresholds primitive is used to set the application quality of service requirements and make it available to the MIHF. These parameters are mapped into media-specific measurements at the MIH layer and then used to configure the link parameter thresholds. While this mapping is not defined by other standards, Table I-1 and Table I-2 provide such mappings. The primitive MIH_Link_Parameters_Report is used to relay link specific measurements back to the MIH User.

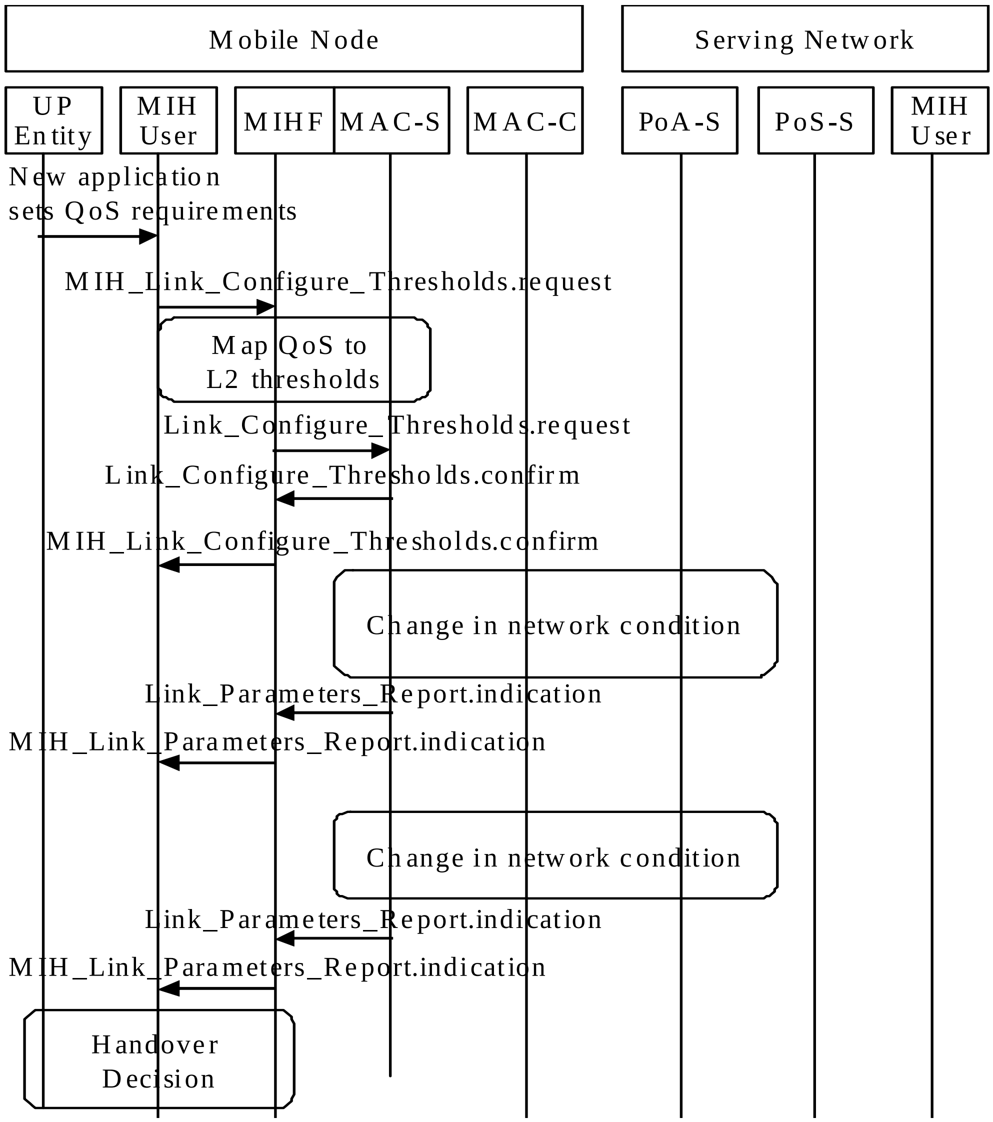


Figure I-1—An example flow for setting application QoS requirements

I.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 I-1 is informative and list measurements defined in IEEE Std 802.11 that may 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 I-1—QoS parameter mapping for IEEE 802.11

802.21 link QoS parameters	Related 802.11 parameters	802.11 IE name	Note
Throughput	Not currently supported.		Measurement is defined as the total number of octets transmitted / Measurement duration.”
Packet error rate	TransmittedFragmentCount MulticastTransmittedFrameCount FailedCount ReceivedFragmentCount * MulticastReceivedFrameCount FCSErrorCount * TransmittedFrameCount RetryCount MultipleRetryCount FrameDuplicateCount RTSSuccessCount RTSFailureCount ACKFailureCount	STA Statistics Report	
Supported number of COS	4 for 802.11e, 8 for HCCA, 1 for non 802.11e systems		
CoS minimum packet transfer delay	Transmit Delay Histogram *	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 *	Transmit Stream/ Category Measurement Report	Trigger (Option) (only to specific STA)
CoS packet delay jitter	Transmit Delay Histogram* Average Transmit Delay*	Transmit Stream/ Category Measurement Report	Trigger(Option) (only to specific STA)

Table I-1—QoS parameter mapping for IEEE 802.11

802.21 link QoS parameters	Related 802.11 parameters	802.11 IE name	Note
CoS packet loss rate	QoSTransmittedFragmentCount * QoSFailedCount QoSRetryCount QoSMultipleRetryCount QoSFrameDuplicateCount QoSRTSSuccessCount QoSRTSFailureCount QoSACKFailureCount * QoSReceivedFragmentCount QoSTransmittedFrameCount QoSDiscardedFrameCount QoSMPDUsReceivedCount QoSRetriesReceivedCount	STA Statistics Report	
	Transmitted MSDU Count * MSDU Discarded Count MSDU Failed Count* MSDU Multiple Retry Count QoS CF-polls Lost Count	Transmit Stream/ Category Measurement Report	Trigger (Option) (only to specific STA)

Note: (*) indicates that the parameters is most likely to be used to directly derive IEEE 802.21 LinkQoSParameters. See I.3 for example derivations.

Table I-2 and Table I-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 I-2—QoS parameter mapping for 802.16 and 3GPP2

802.21 link QoS parameters	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 I-3—QoS parameter mapping for 3GPP

802.21 link QoS parameters	Related 3GPP parameters			
Supported number of CoS	4			
	Conversational	**Streaming**	**Interactive**	**Background**
Throughput	Peak throughput			
	Mean throughput			
	Maximum bit rate for uplink/downlink			
	Guaranteed bit rate for uplink/downlink			
Link packet error rate	SDU Error Ratio			
	Residual Bit Error Rate			
CoS minimum packet transfer delay	Transfer delay			
CoS average packet transfer delay	Transfer delay			
CoS maximum packet transfer delay	Maximum Transfer delay			
CoS packet transfer delay jitter		Delay Variation		
CoS packet loss rate	Residual Bit Error Rate			
	SDU Error Ratio			

I.3 Deriving generic IEEE 802.21 QoS parameters

I.3.1 General

The following subclauses describes how to derive generic QoS parameters from IEEE Std 802.11 link measurement parameters. This derivation relies on incremental values of counters as specified in the IEEE Std. 802.11.

Note that the parameters are unicast parameters that are unrelated to multicast traffic.

I.3.2 Packet loss rate

To calculate the packet loss rate (PLR), one uses the following equation.

$$PLR = \frac{\text{the number of lost packets}}{\text{the number of transmitted packet (successful + failed)}}$$

According to IEEE Std 802.11, a packet is a MAC user data packet or MAC service data unit (MSDU).

The PLR_{MSDU} can be derived from the **QoS Metric information element** using the following equation.

$$PLR_{MSDU} = \frac{\text{failed MSDUs}}{\text{Transmitted MSDUs} + \text{Failed MSDUs}} = \frac{\text{MSDUFailedCount}}{\text{TransmittedMSDUCount} + \text{MSDUFailedCount}}$$

I.3.3 Packet error rate

The packet error rate (PER) can be calculated using the following equation.

$$PER = \frac{\text{the number of packets that are received with errors}}{\text{the number of packets in a population of interest}}$$

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 the following equation.

$$PER = \frac{\text{FCSErrorCount}}{\text{ReceivedFragmentCount} + \text{FCSErrorCount}}$$

I.3.4 Average transfer delay

In 802.11k, the transmit delay (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 **Transmit Stream/Category Measurement Report.**

$$ATD_{MSDU} = \text{Average MSDU Transmit Delay} = \text{Average Transmit Delay}$$

I.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:
  - Transmit Delay Histogram and
  - Average Transmit Delay parameters

Variance calculation using discrete density function is given as

$$VAR(X) = \sum_{i=1}^{N} P_i \left(x_i - \overline{X}\right)^2$$

Therefore, the packet transfer delay jitter for MSDU level is

$$\text{Packet Transfer Delay Jitter} = \text{MSDU Packet Transmit Delay Jitter}$$
$$= \sqrt{\sum_{i=1}^{N} P_i \left(x_i - \text{AverageTransmitDelay}\right)^2}$$

where,

N= the number of bins of Transmit Delay Histogram,

P_i= the value (measured percentile) of i-th bin of Transmit Delay Histogram, and

x_i=the mean value of the delay range of i-th bin.

Annex J Media specific mapping for SAPs

(normative)

The MIHF aggregates disparate interfaces with respective media dependent lower-layer instances (media dependent service access points) into a single interface with the MIH Users (the MIH SAP), reducing the inter-media differences to the extent possible.

The MIHF 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 MIHF 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 MIHF capabilities.

The following subclauses list general amendments recommended to different underlying access technology standards due to the enhanced heterogeneous handover capability provided by MIHF.

J.1 MIH_LINK_SAP mapping to specific technologies

Table J-1—MIH_Link_SAP/802.16 primitives mapping

MIH_LINK_SAP primitive	IEEE 802.16 C_SAP	IEEE 802.16 M_SAP
Link_Detected	C-HO-RSP (HO-Scan)	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_Handover_Imminent	N/A	N/A
Link_Handover_Complete	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

Table J-1—MIH_Link_SAP/802.16 primitives mapping

MIH_LINK_SAP primitive		IEEE 802.16 C_SAP	IEEE 802.16 M_SAP
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 J-2—MIH_LINK_SAP/802.11/802.3/802.1ag primitives mapping

Primitives	IEEE 802.11	IEEE 802.3	IEEE 802.1ag
Link_Detected	N/A	N/A	N/A
Link_Up	MLME-LinkUp.indication	Link fault	dot1agCfgFaultAlarm*
Link_Down	MLME-LinkDown.indication	Link fault	dot1agCfgFaultAlarm*
Link_Parameters_Report	MLME-MEASURE.confirm MLME-MREPORT.indication†	N/A	N/A
Link_Going_Down	MLME-LinkGoingDown.indication	Dying Gasp	N/A
Link_Handover_Imminent	MLME-LinkHandoverImminent.indication	N/A	N/A
Link_Handover_Complete	MLME-LinkHandoverComplete.indication	N/A	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	N/A	N/A	N/A
Link_Configure_Thresholds	MLME-MEASURE.request MLME-MREQUEST.request‡	N/A	N/A
Link_Action	N/A	N/A	N/A

*The alarms (cross-connection, link failure, MACstatusDefect, and RDIdefect) are enabled and no other higher priority event has occurerd.

†IEEE 802.11k 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 MIHF 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.

‡It is used to configure threshold values for Link_Parameters_Report. Thresholds are used for triggering reports. IEEE 802.11k 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_Threshold.confirm. MLME-MEASURE.confirm is used to deliver the measurement results not to confirm the threshold setting.

Table J-3—MIH_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_Handover_Imminent	N/A	N/A
Link_Handover_Complete	RABMAS-RAB-ESTABLISH RABMSM-MODIFY	L2.Data.Confirm
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

J.2 Mappings from MIH_LINK_SAP to media-specific SAPs

J.2.1 802.3

LSAP, defined in the IEEE Std 802.2, provides the interface between the MIHF and the Logical Link Control sublayer in IEEE 802.3 network. This SAP is used for local MIH exchanges between the MIHF and the lower layers of the IEEE 802.3 interface (as the IEEE 802.3 instantiation of the MIH_LINK_SAP) and for the L2 transport of MIH messages across IEEE 802.3 access links.

J.2.2 802.11

The MIHF uses MSGCF_SAP for interfacing with the link layer of IEEE 802.11 networks. The MIH_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 P802.11u/D2.0 and it includes, but is not limited to primitives related to:

- System configuration
- Link state change notifications/triggers
- MIH frame transport through control or management frames

LSAP, defined in the IEEE Std 802.2, provides the interface between the MIHF and the Logical Link Control sublayer in IEEE 802.11. This SAP is used for the L2 transport of MIH messages across IEEE 802.11 access links. The MIH messages are carried in 802.11 data frames.

Table J-2 lists this mapping.

J.2.3 802.16

The MIHF 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 802.16g and it includes primitives related to:

- Handovers (e.g., notification of HO 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., EAP payloads)
- Media Independent Function Services

M_SAP is defined by IEEE Std 802.16g and it includes primitives related to:

- System configuration
- Monitoring Statistics
- Notifications Triggers
- Multi-mode interface management

CS_SAP, defined in the IEEE Std 802.16, provides the interface between the MIHF and the service-specific Convergence Sublayer in IEEE 802.16 networks. This SAP is used for the L2 transport of MIH messages through data frames across IEEE 802.16 access links.

Table J-1 lists this mapping.

J.2.4 3GPP and 3GPP2

This SAP defines MIH_3GLINK_SAP interface between the MIHF and the different protocol elements of the 3G system.

3GPP and 3GPP2 service primitives for GERAN, UMTS, long term evolution (LTE), cdma2000, cdma2000-HRPD and UMB are used to access MIH services. This is done by establishing a relationship between the 3GPP/3GPP2 primitives and MIH primitives.

Table J-3 lists this mapping. Note that a 3GPP primitive group can be mapped to more than one MIH primitive, as shown in Table J-3.

Annex K Making user extensions to MIIS schema

(informative)

This annex describes how to create an extended schema. How to create “IP address of Mobile IP home agent” is used as an example.

It is possible to support an Extended Schema without defining additional IEs or TLVs (including vendor specific ones) other than those that are defined in this standard. An extended schema can be defined as an XML document. An example Extended Schema definition is shown as follows.

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
 <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
 <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
 <!ENTITY mihbasic "URL_TO_BE_ASSIGNED">
 <!ENTITY mihextended "http://www.informationservice.org/2006/08/extended-schema#">
 <!ENTITY owl "http://www.w3.org/2002/07/owl#">
 <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>

<rdf:RDF xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;" xmlns:mihbasic="&mihbasic;"
xmlns:mihextended="&mihextended;" xml:base="&mihextended;" xmlns:owl="&owl;" xmlns:xsd="&xsd;">

 <owl:Ontology rdf:about="">
  <rdfs:label>Extended Schema</rdfs:label>
 </owl:Ontology>

<owl:ObjectProperty rdf:ID="ha_address">
  <rdfs:domain rdf:resource="&mihbasic;NETWORK"/>
  <rdfs:range rdf:resource="&mihbasic;TRANSPORT_ADDR"/>
 </owl:ObjectProperty>

</rdf:RDF>
```

Annex L Handover procedures

(informative)

L.1 Mobile-initiated handover procedure

The Mobile-initiated Handover Procedure operates as follows (see Figure L-1a, Figure L-1b, and Figure L-1c):

1) Mobile Node is connected to the serving network via Current PoS and it has access to MIH Information Server.

2) Mobile Node queries information about neighboring networks by sending the MIH_Get_Information Request to Information Server. Information Server responds with MIH_Get_Information Response. This information query is attempted as soon as Mobile Node is first attached to the network.

3) Mobile Node triggers a mobile-initiated handover by sending MIH_MN_HO_Candidate_Query Request to Serving PoS. This request contains the information of potential candidate networks.

4) Serving PoS queries the availability of resources at the candidate networks by sending MIH_N2N_HO_Query_Resources Request to one or multiple Candidate PoSs.

5) Candidate PoSs respond with MIH_N2N_HO_Query_Resources Response and Serving PoS notifies the Mobile Node of the resulting resource availability at the candidate networks through MIH_MN_HO_Candidate_Query Response.

6) Mobile Node decides the target of the handover and notifies the Serving PoS of the decided target network information by sending the MIH_MN_HO_Commit request message. Also, the Mobile Node commits a link switch to the target network interface by invoking the MIH_Link_Actions.request primitive.

7) Serving PoS sends the MIH_N2N_HO_Commit Request to Target PoS to request resource preparation at the target network. Target PoS responds the result of the resource preparation by MIH_N2N_HO_Commit Response.

8) The new layer 2 connection is established and a certain mobility management protocol procedures are carried out between Mobile Node and target network.

9) Mobile Node sends MIH_MN_HO_Complete Request to Target PoS. Target PoS sends MIH_N2N_HO_Complete Request to previous Serving PoS to release resource, which was allocated to Mobile Node. After identifying that resource is successfully released, Target PoS sends MIH_MN_HO_Complete Response to Mobile Node.

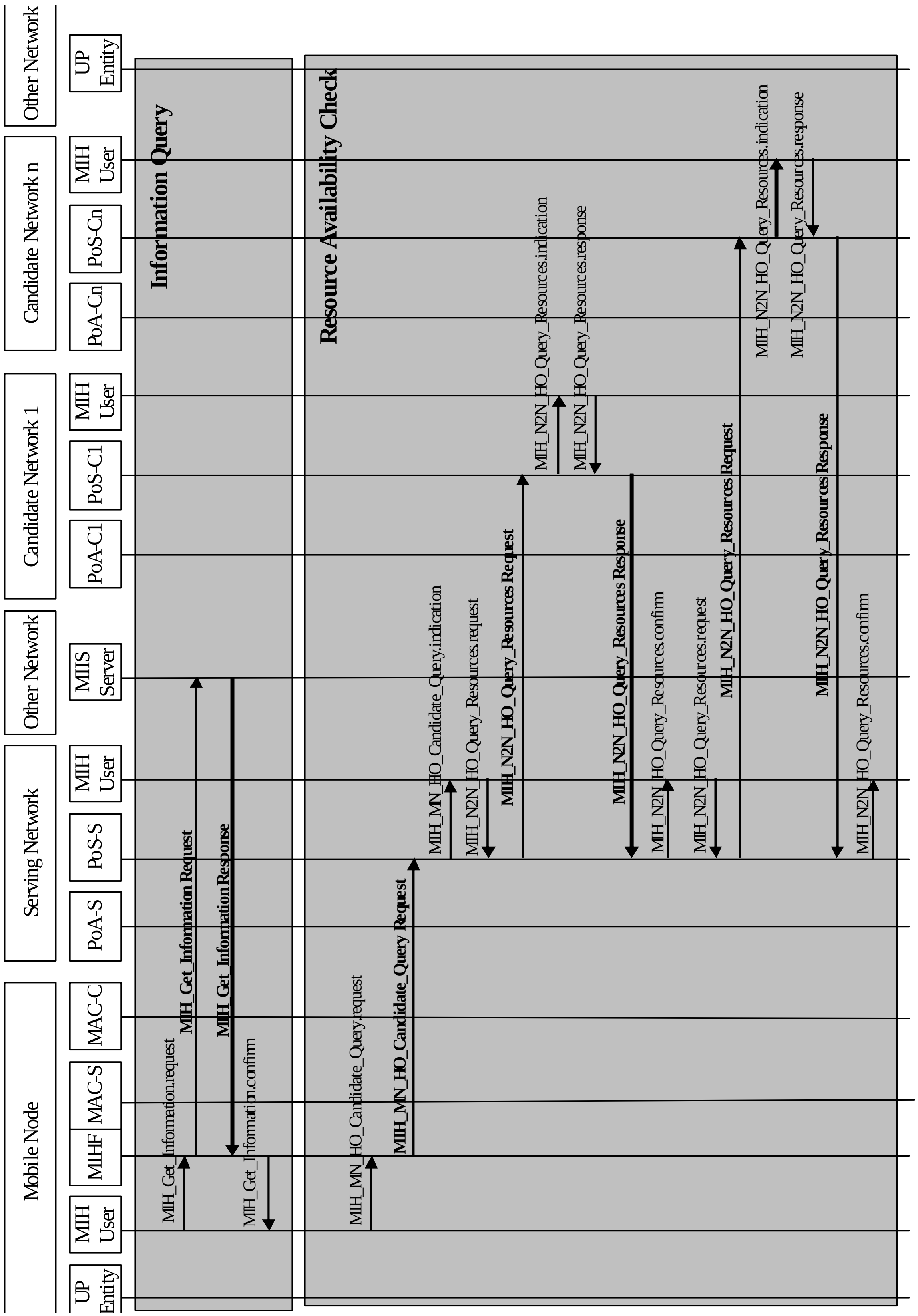


Figure L-1—Mobile-initiated handover procedure

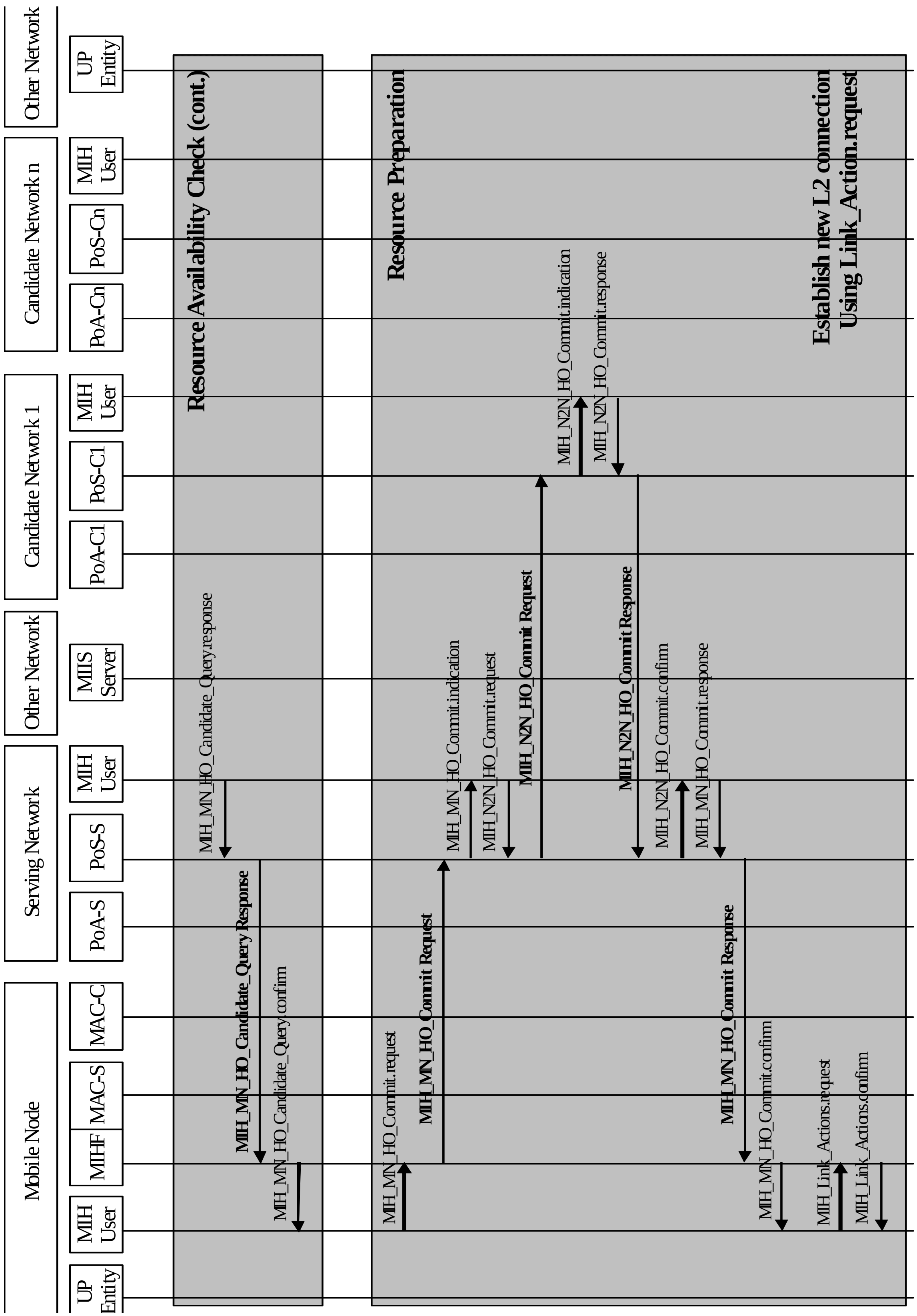


Figure L-1a—Mobile-initiated handover procedure (cont.)

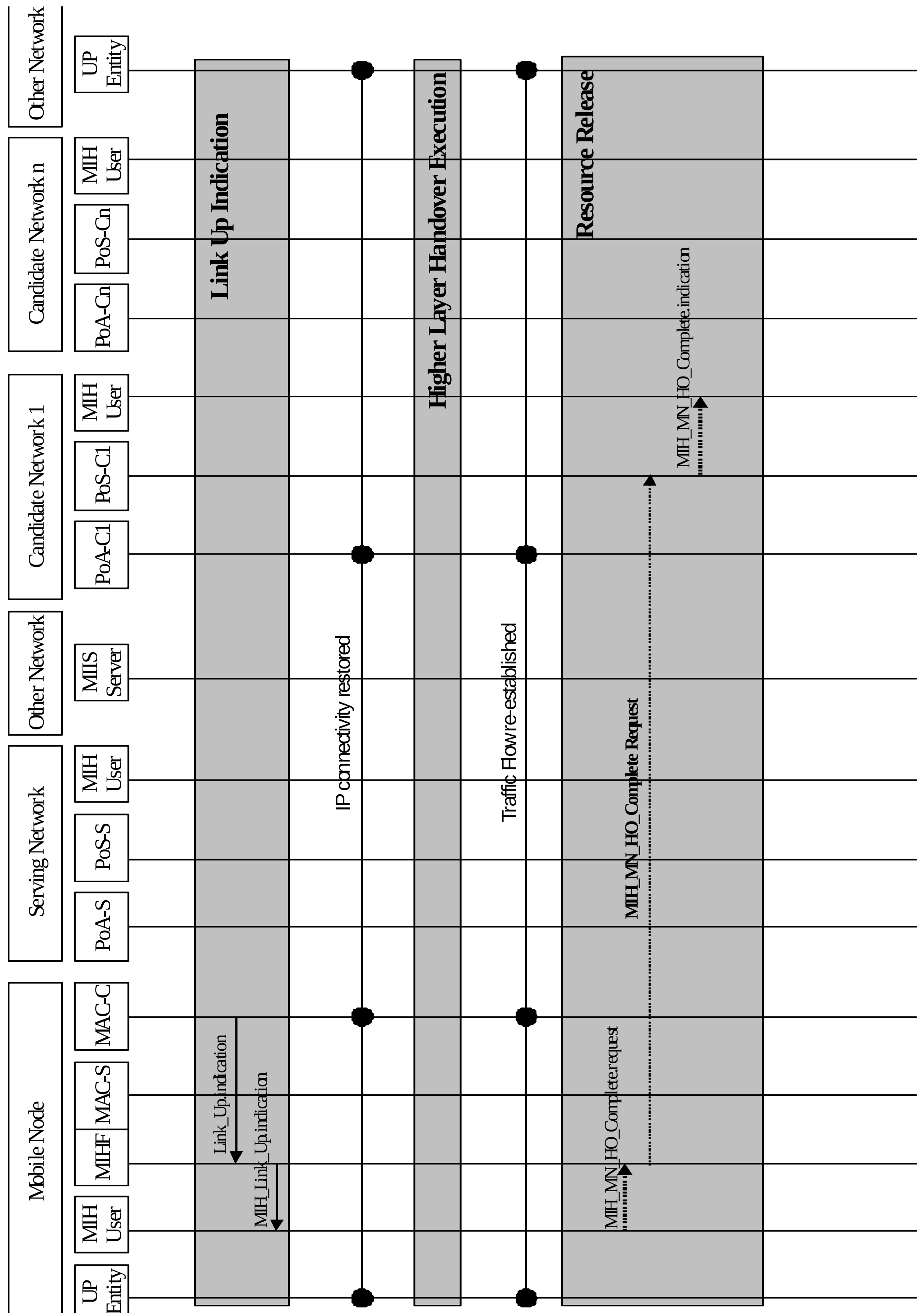


Figure L-1b—Mobile-initiated handover procedure (cont.)

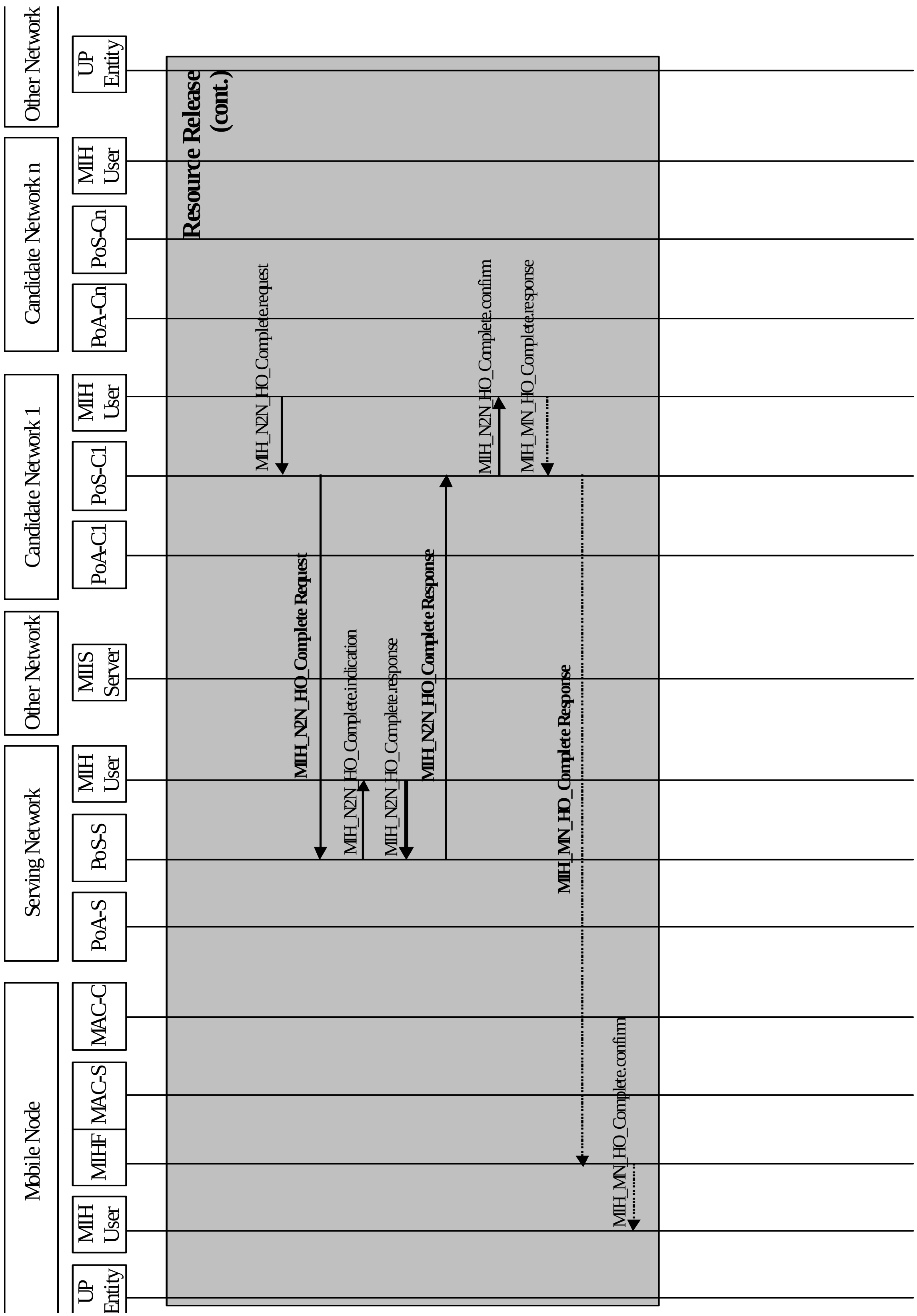


Figure L-1c—Mobile-initiated handover procedure (cont.)

L.2 Network-initiated handover procedure

The Network-initiated Handover Procedure operates as follows (see Figure L-2, Figure L-2a, Figure L-2b, and Figure L-2c):

1) Serving PoS sends MIH_Get_Information Request to Information Server to get neighboring network information and Information Server responds by sending MIH_Get_Information Response.

2) Serving PoS triggers a network-initiated handover by sending MIH_Net_HO_Candidate_Query Request to Mobile Node. The MN responds through MIH_Net_HO_Candidate_Query Response, which contains Mobile Node's acknowledgement about the handover and its preferred link and PoS lists.

3) Serving PoS sends MIH_N2N_HO_Query_Resources Request to one or more Candidate PoSs to check the availability of the resource at candidate networks. Candidate PoS responds by sending MIH_N2N_HO_Query_Resources Response to Serving PoS.

4) Serving PoS decides the target of the handover based on the available resource status at candidate networks.

5) Serving PoS sends MIH_N2N_HO_Commit Request to Target PoS to prepare resource at the target network. Target PoS responds the result of the resource preparation by sending MIH_N2N_HO_Commit Response.

6) After identifying that resource is successfully prepared, Serving PoS commands Mobile Node to commit handover toward the specified network type and PoA through MIH_Net_HO_Commit Request.

7) New layer 2 connection is established and Mobile Node sends MIH_Net_HO_Commit Response to Serving PoS.

8) After higher layer handover execution, Mobile Node sends MIH_MN_HO_Complete Request to Target PoS. Target PoS sends MIH_N2N_HO_Complete Request to previous Serving PoS to release resource, which was allocated to Mobile Node. After identifying that resource is successfully released, Target PoS sends MIH_MN_HO_Complete Response to Mobile Node.

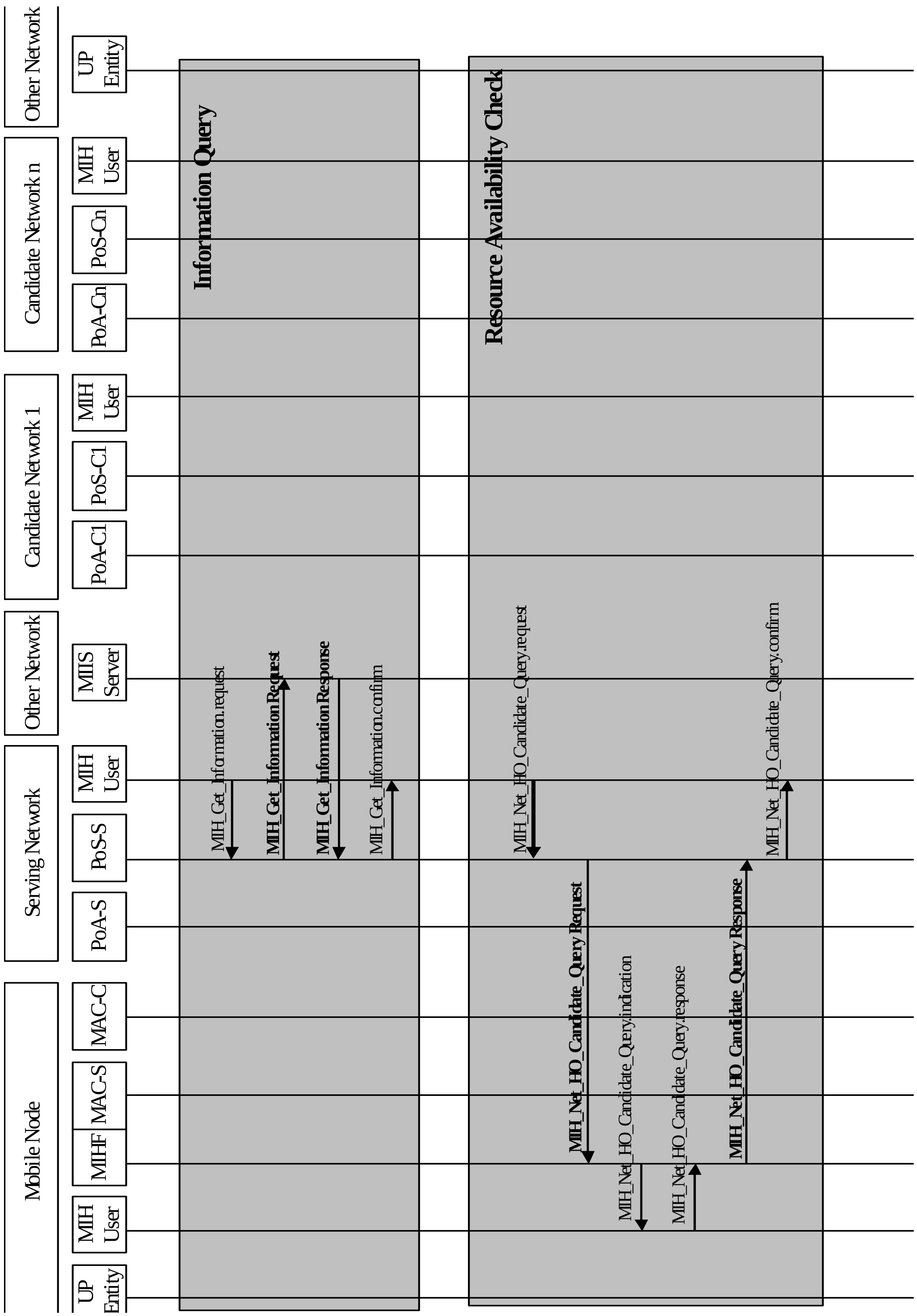


Figure L-2—Network-initiated handover procedure

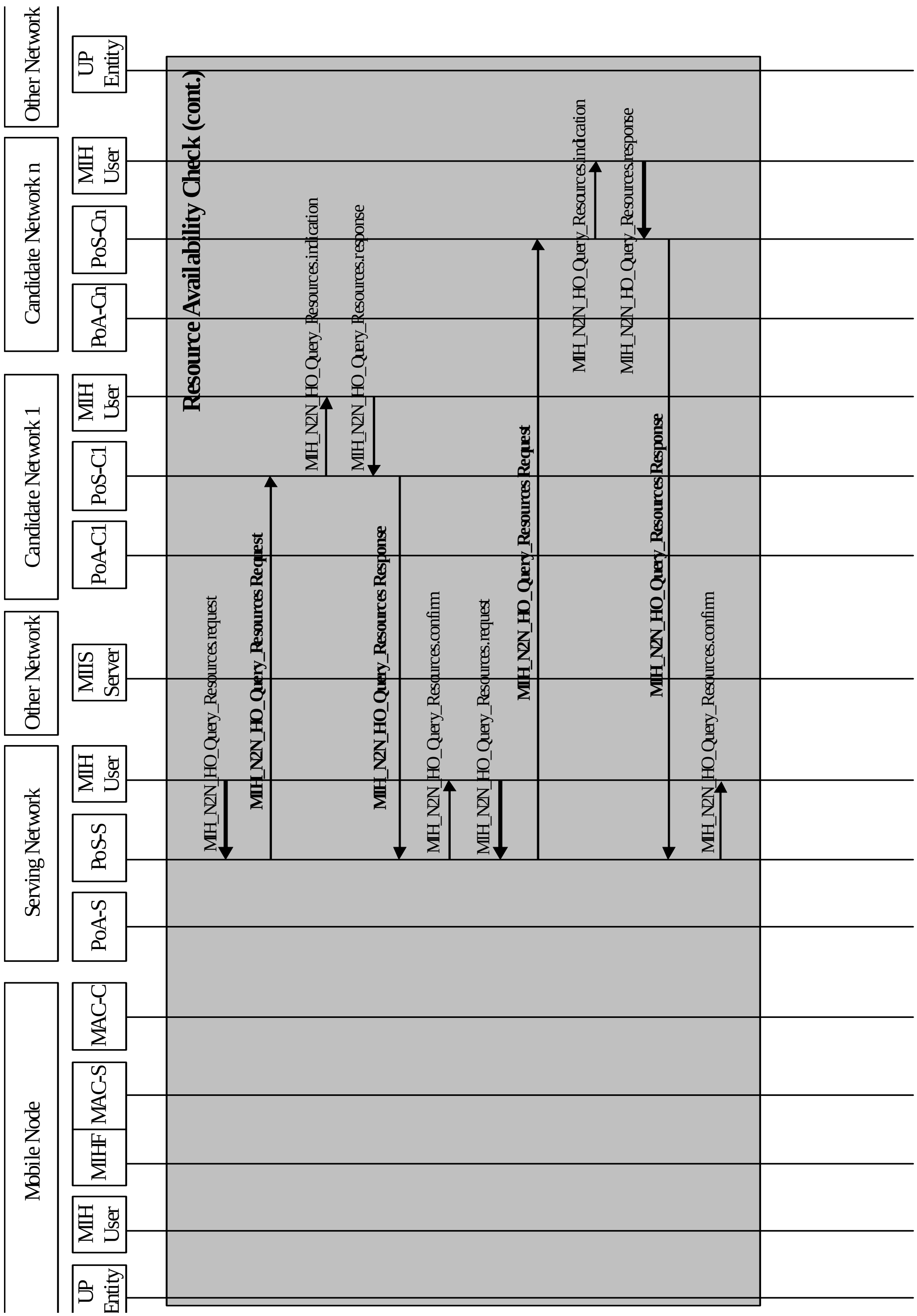


Figure L-2a—Network-initiated handover procedure (cont.)

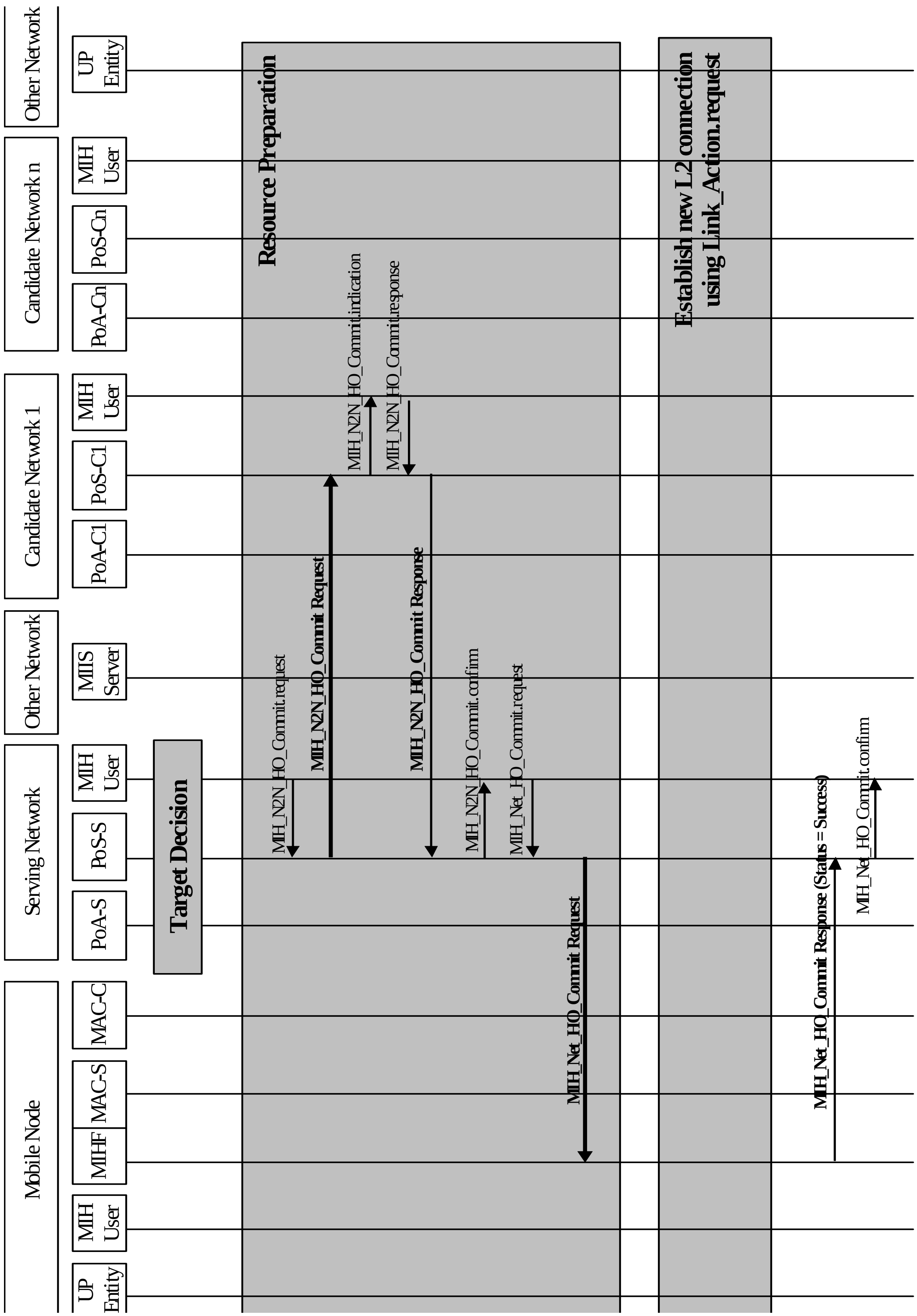


Figure L-2b—Network-initiated handover procedure (cont.)

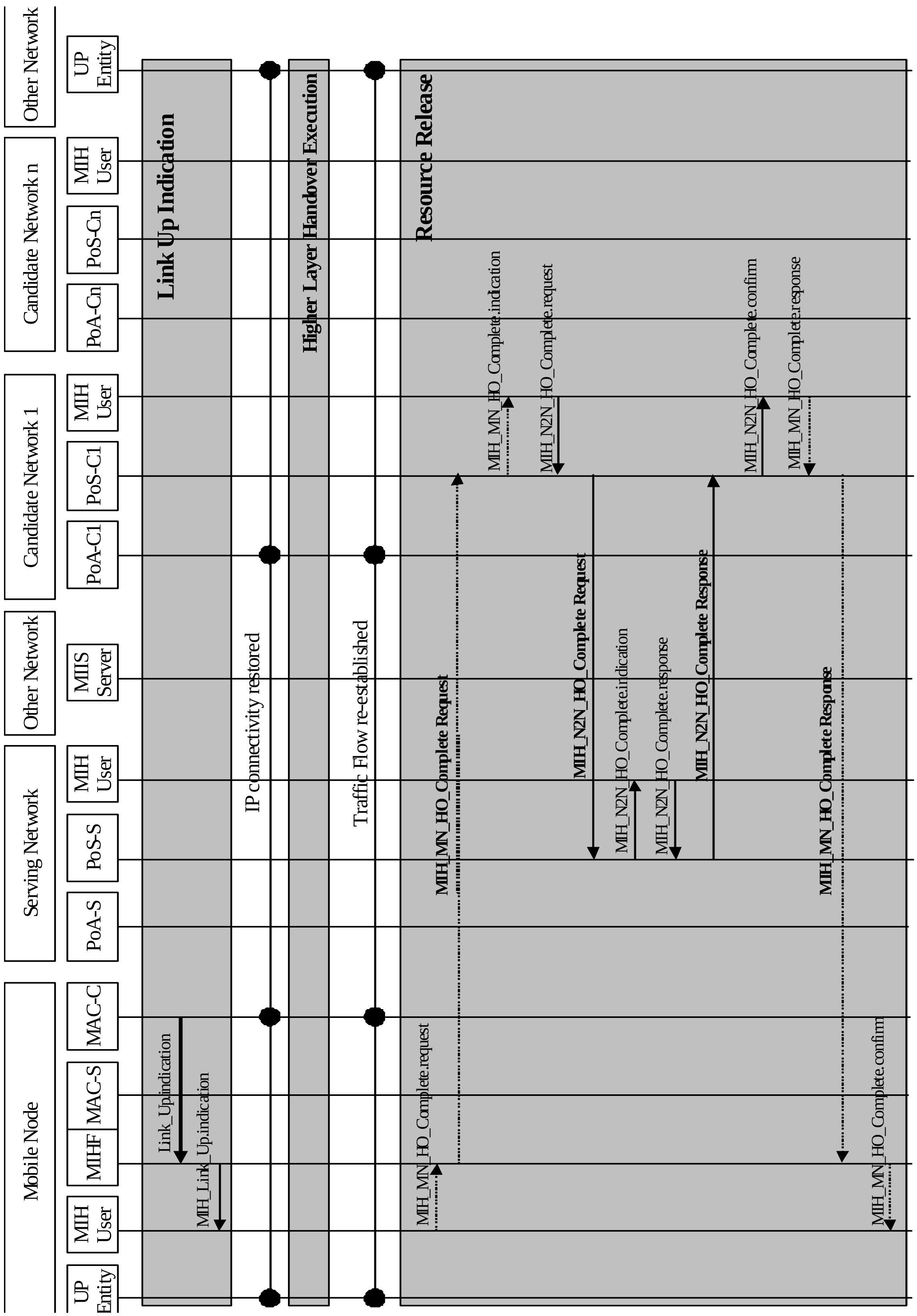


Figure L-2c—Network-initiated handover procedure (cont.)

L.3 Example handover flow chart between 802.11 and 802.16

Figure L-3 shows a handover flow chart between the 802.11 and the 802.16 network. This is an example of dual radio handover procedure wherein both the radios involved in handover can transmit and receive at the same time. The handover procedure operates as follows:

1) The Mobile Node is connected to the 802.11 network and receives the 802.11 link measurement report through the MIH_Link_Parameters_Report.indication and acquires the neighboring network information by the MIH_Get_Information.confirm.

2) When the Link_Going_Down event happens on the current 802.11 network, the Mobile Node performs the MIH_Link_Actions.request to scan the link status of the candidate networks. The mobile node discovers the 802.16 network and can acquire the candidate 802.16 network's DL_MAP, UL_MAP, DCD and UCD parameters.

3) The Mobile Node identifies the resource availability status of the candidate network by sending the MIH_MN_HO_Candidate_Query message to the Serving PoS. When the Serving PoS receives the MIH_MN_HO_Candidate_Query Request from the Mobile Node, it retrieves resource information from target network by sending MIH_N2N_HO_Query_Resources message to the PoSs on the candidate networks.

4) Based on resource availability and other selection criteria the 802.16 network is selected as the target of the handover and the Mobile Node sends MIH_MN_HO_Commit request message to the Serving PoS to notify the decided target network information. The Serving PoS reserves the resource at the target network through MIH_N2N_HO_Commit messages.

5) The Mobile Node commits a link switch to the802.16 interface and the new layer 2 connection for the target 802.16 network is established. The Mobile IP procedures are carried out between the Mobile Node and the 802.16 network. As a result of that, the active sessions are now shifted over to the 802.16 network.

6) The Mobile Node sends the MIH_MN_HO_Complete Request to the Serving PoS on the 802.16 network and that Serving PoS exchanges the MIH_N2N_HO_Complete messages with the previous PoS on the 802.11 network to release the resource that was reserved for the Mobile Node on that network.

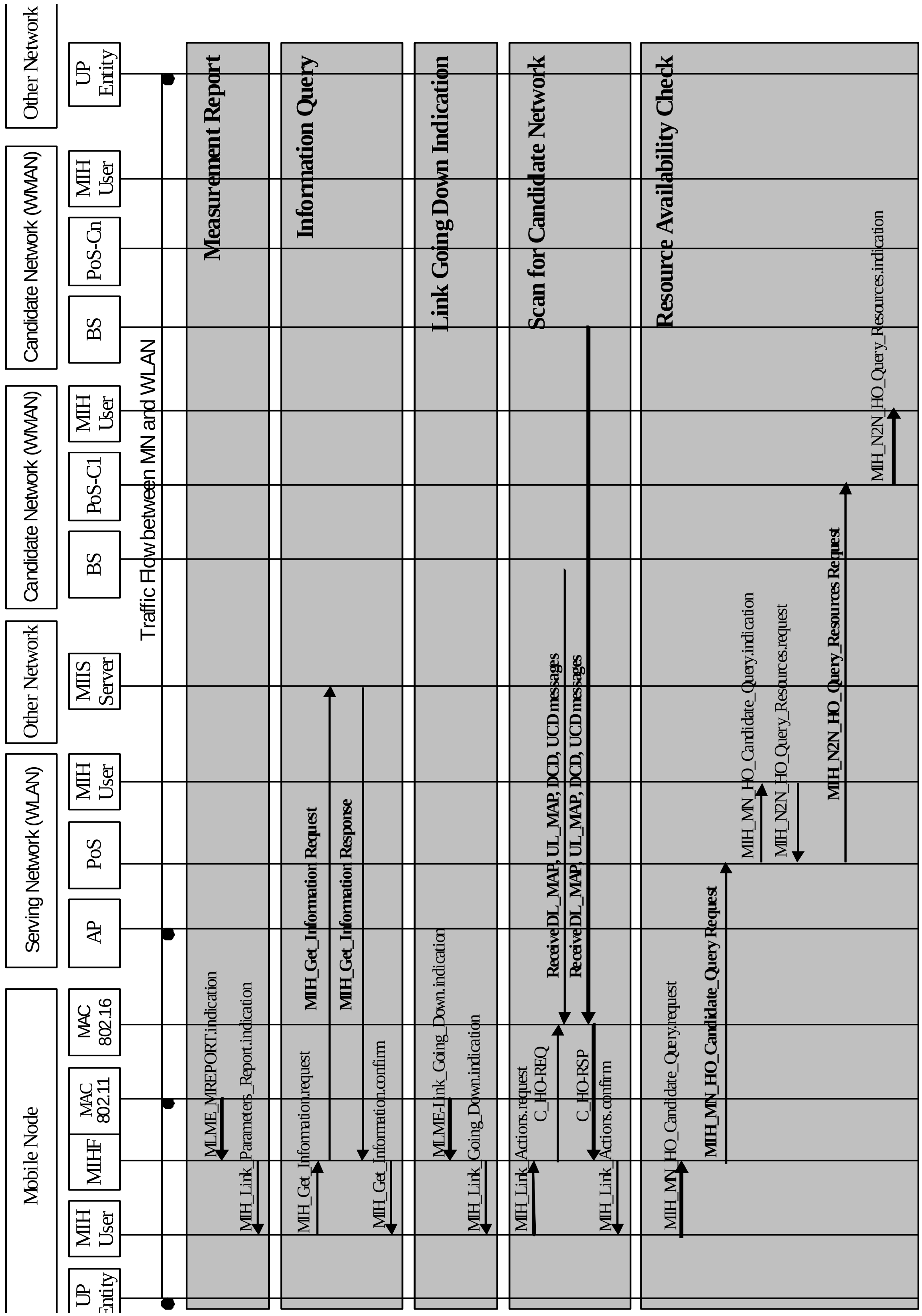


Figure L-3—Example handover flow chart between 802.11 and 802.16

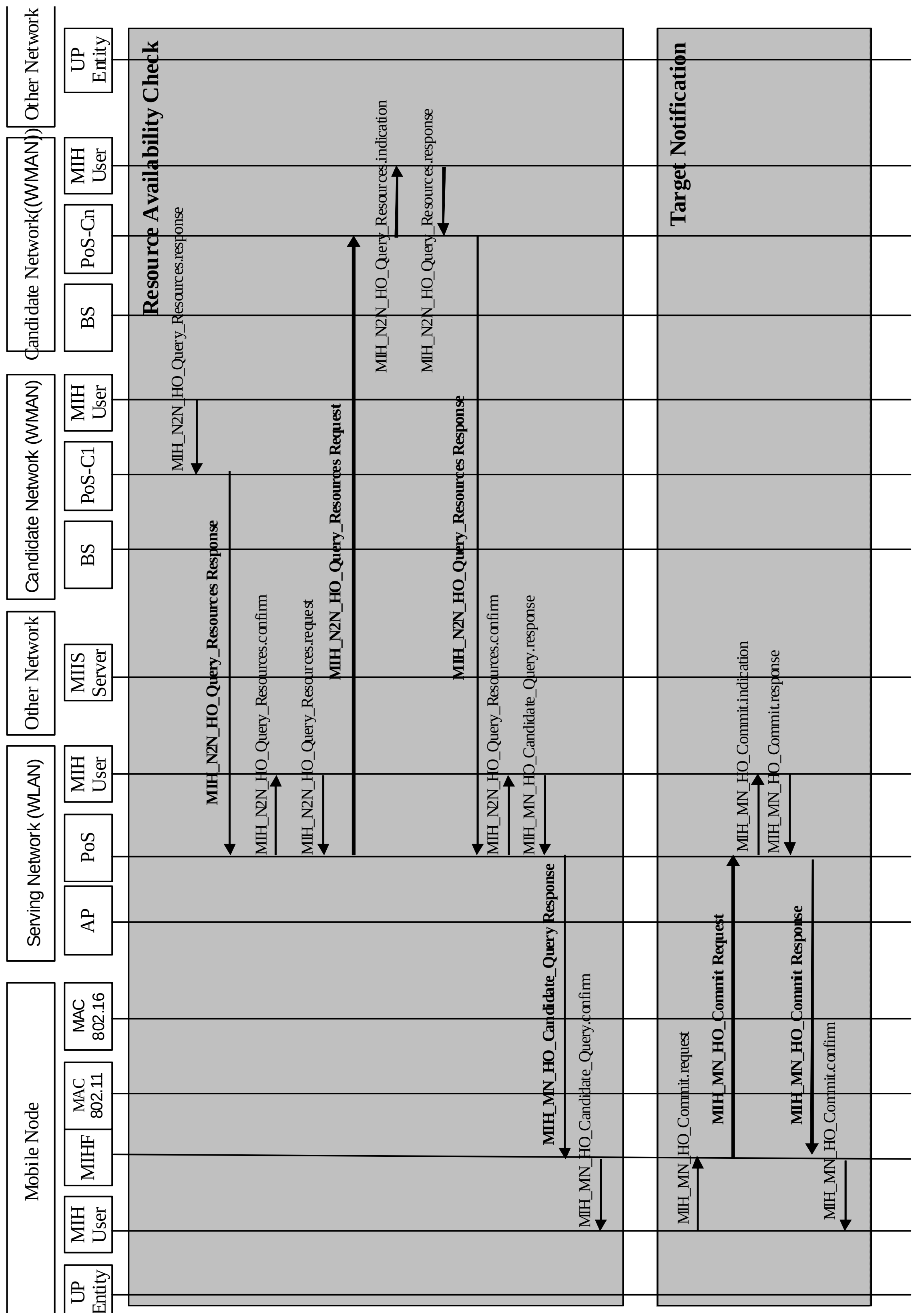


Figure L-3a—Example handover flow chart between 802.11 and 802.16 (cont.)

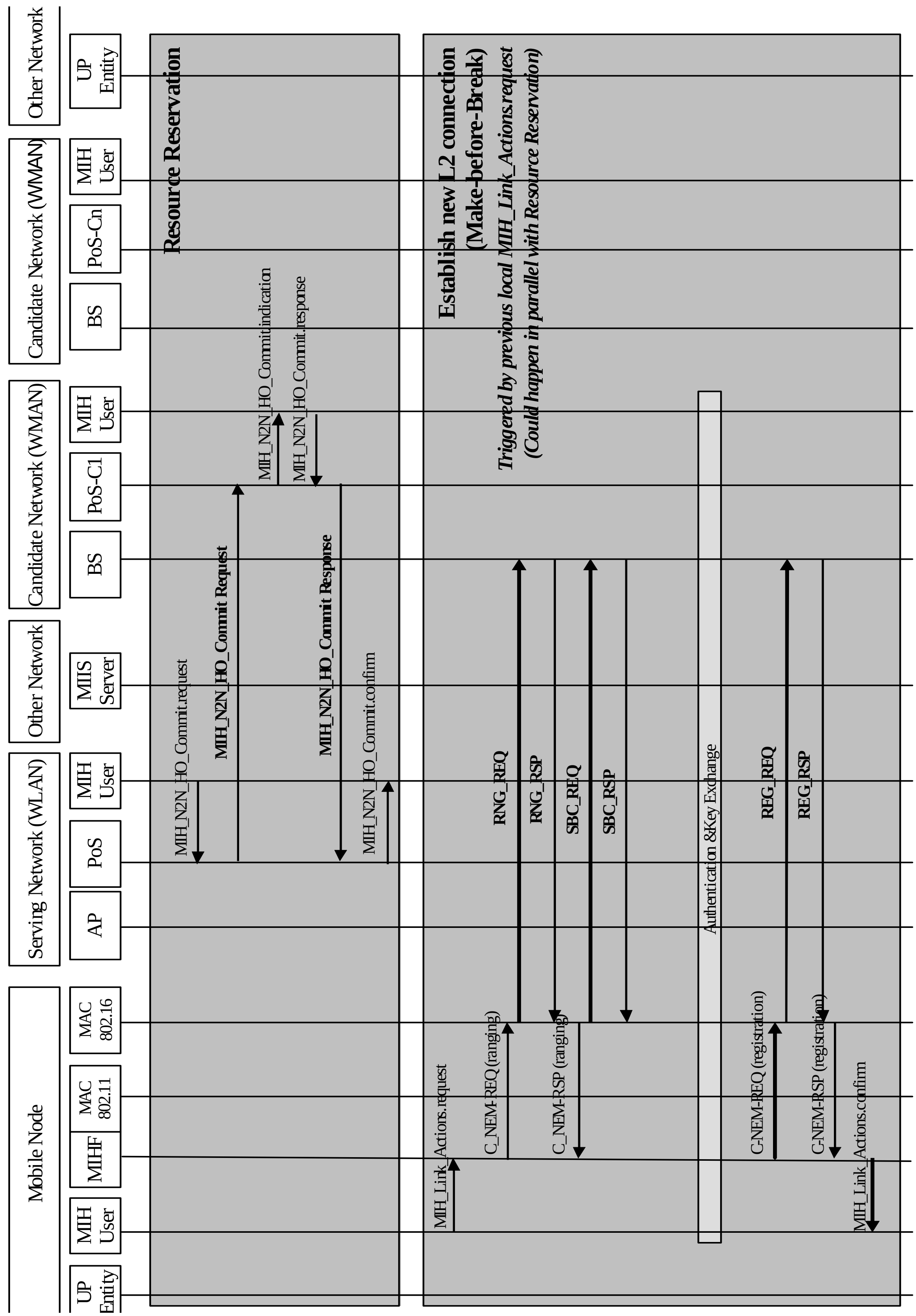


Figure L-3b—Example handover flow chart between 802.11 and 802.16 (cont.)

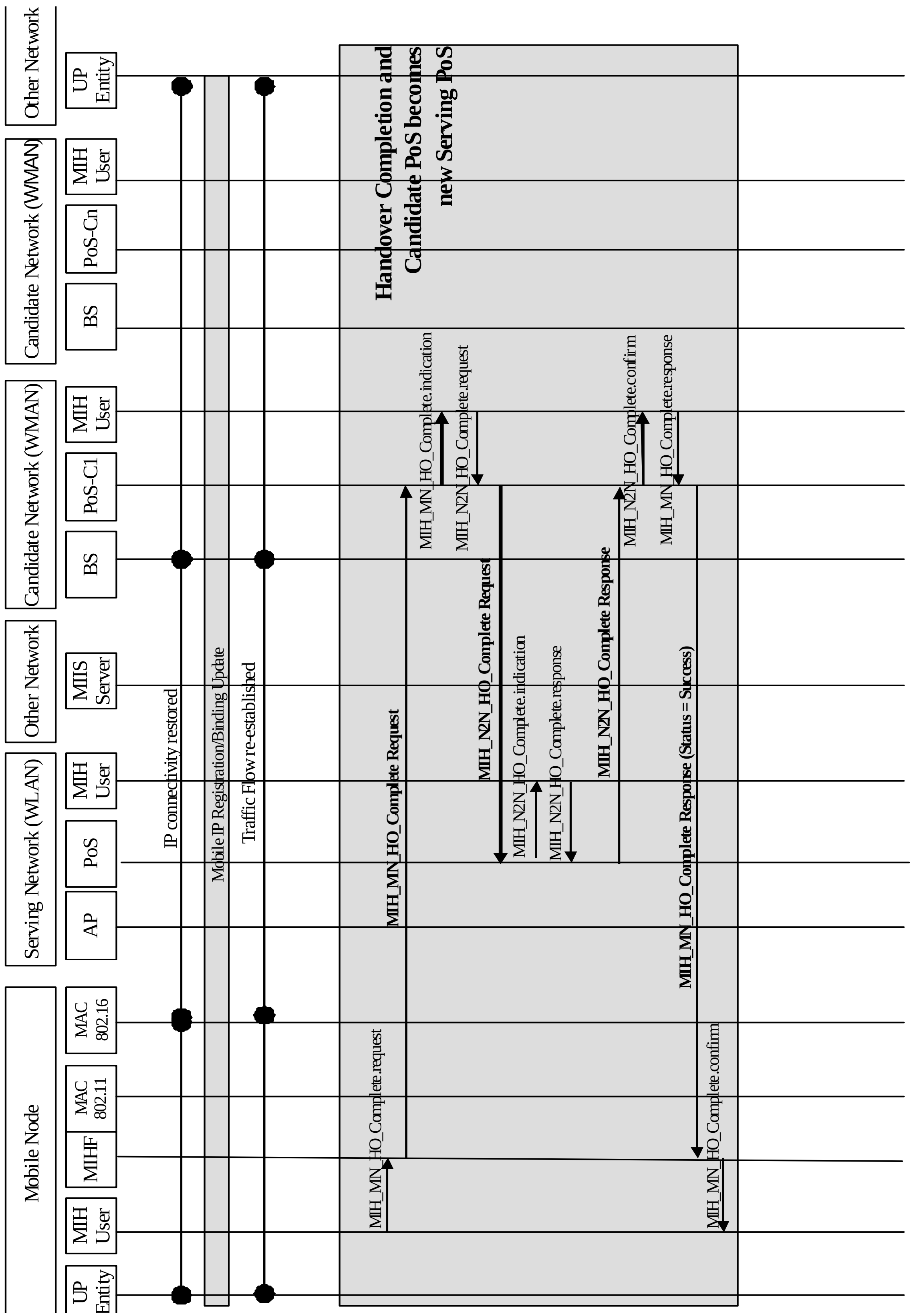


Figure L-3c—Example handover flow chart between 802.11 and 802.16 (cont.)

L.4 Example handover flow chart for Proxy Mobile IPv6

L.4.1 Network-initiated handover procedures

Figure L-4 shows a network-initiated handover flow chart for Proxy Mobile IPv6 (PMIPv6), which is currently under standardization for supporting a local mobility in IETF NetLMM Working Group (Although the Proxy Mobile IP is under standardization, its overall flow is already defined. The following handover flow refers to the overall flow). The handover flow operates as follows:

1) MN receives packets through both Mobile Access Gateway (MAG) 1 located in the serving network and Local Mobility Anchor (LMA), which are primary components of the PMIPv6.

2) The Serving PoS queries the Information Server to get information about available neighboring networks.

3) The Serving PoS triggers a network-initiated handover by sending the MIH_Net_HO_Candidate_Query Request message to the MN. The MN responds through the MIH_Net_HO_Candidate_Query Response message, which contains MN's acknowledgement about the handover initiation and its preferred link and PoS lists.

4) The Serving PoS sends the MIH_N2N_HO_Query_Resource Request messages to different Candidate PoSs (can be more than one) to query the availability of the resource at candidate networks. The Candidate PoSs respond by sending the MIH_N2N_HO_Query_Resource Response message to the Serving PoS. The Serving PoS decides the handover target based on the resource availability information of candidate networks informed by the MIH_N2N_HO_Query_Resource Response message.

5) The Serving PoS informs the decided Target PoS (i.e. Candidate Network 1 in the Figure L-4, where MAG2 is located) of the handover commitment and requests the Target PoS to prepare resources for the incoming MN through sending the MIH_N2N_HO_Commit Request message. The Target PoS replies the result of the handover commitment and resource preparation by sending MIH_N2N_HO_Commit Response. (Upon receiving the MIH_N2N_HO_Commit Request message, PMIPv6 client in the Target PoS queries the incoming MN's profile to an AAA server and send Proxy Binding Update in order to register the location of the MN in advance. The PMIPv6 client in the Target PoS buffers the packets received from LMA until the MN attaches to the Target PoS.)

6) The Serving PoS requests MN to perform handover to the decided Target PoS by sending the MIH_Net_HO_Commit Request message. The MN replies the result of the handover commitment by sending MIH_Net_HO_Commit Response message.

7) Upon detecting MN's detachment, PMIPv6 client in the Serving PoS terminates a current binding of the MN via sending Proxy Binding Update with Lifetime set to 0 and requests LMA to buffer packets destined for the MN.

8) Once the MN establishes Layer 2 connection to the Target PoS, PMIPv6 client in the Target PoS registers the current MN's location to LMA by sending Proxy Binding Update message. LMA updates its Binding Cache Entry with the Proxy Binding Update message and then replies with Proxy Binding Acknowledgement message. LMA forwards the buffered packets.

9) After receiving the Proxy Binding Acknowledgement message, PMIPv6 clients sends Router Advertisement message to the MN. The Router Advertisement is constructed with the MN's information obtained from the policy server and LMA. It can be solicited by Router Solicitation message from the MN or periodically transmitted. MN configures IP addresses on its interface, which is currently used to connect to the Target PoS, with the received Router Advertisement message. Once the PMIPv6 procedures are completed, MN receives packets through both MAG 2 and LMA.

10) After the PMIPv6 execution, the Target PoS sends the MIH_N2N_HO_Complete Request message to the previous Serving PoS. The previous Serving PoS responds the message with MIH_N2N_HO_Complete Response.

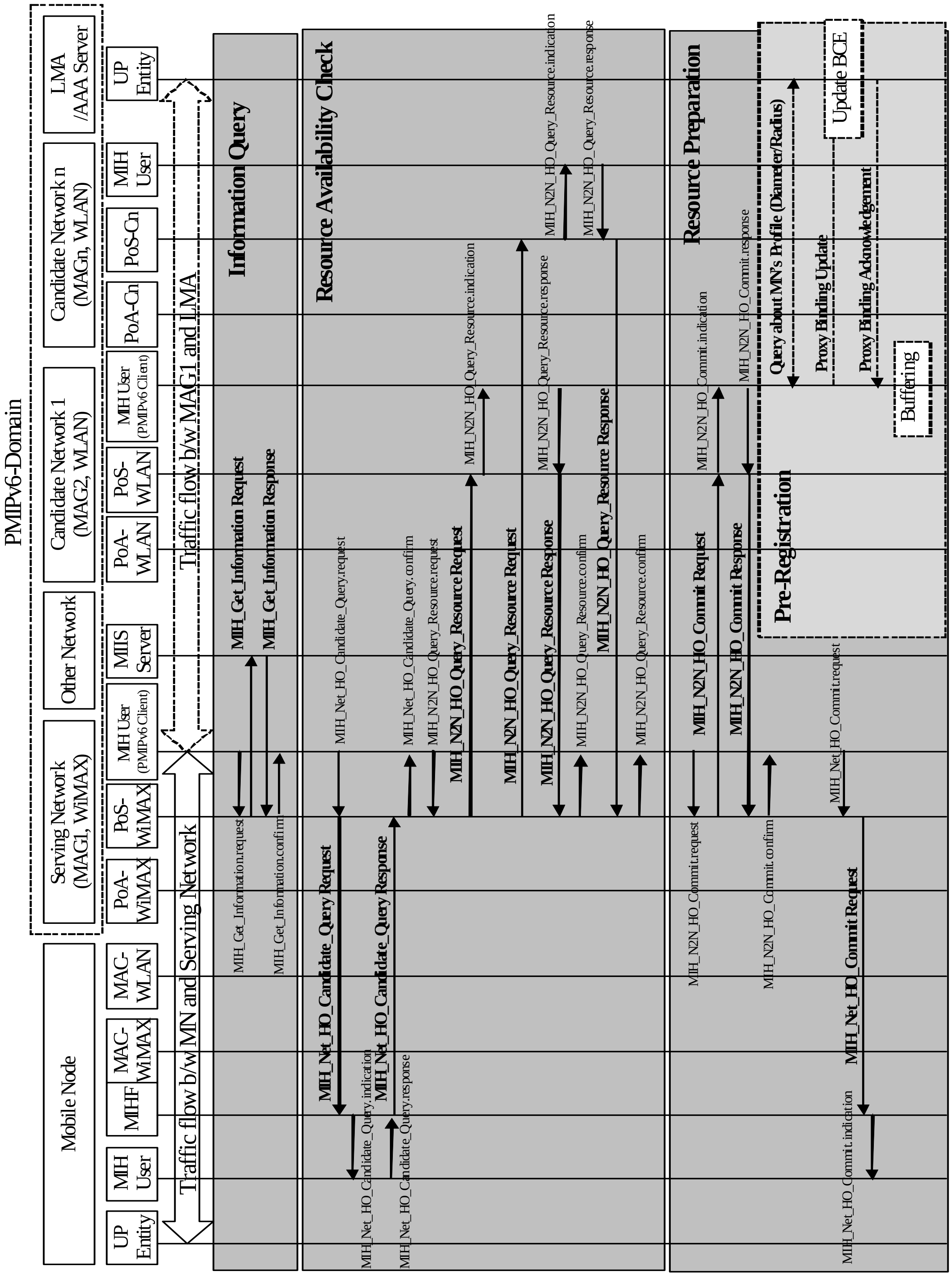


Figure L-4—Network-initiated handover procedures

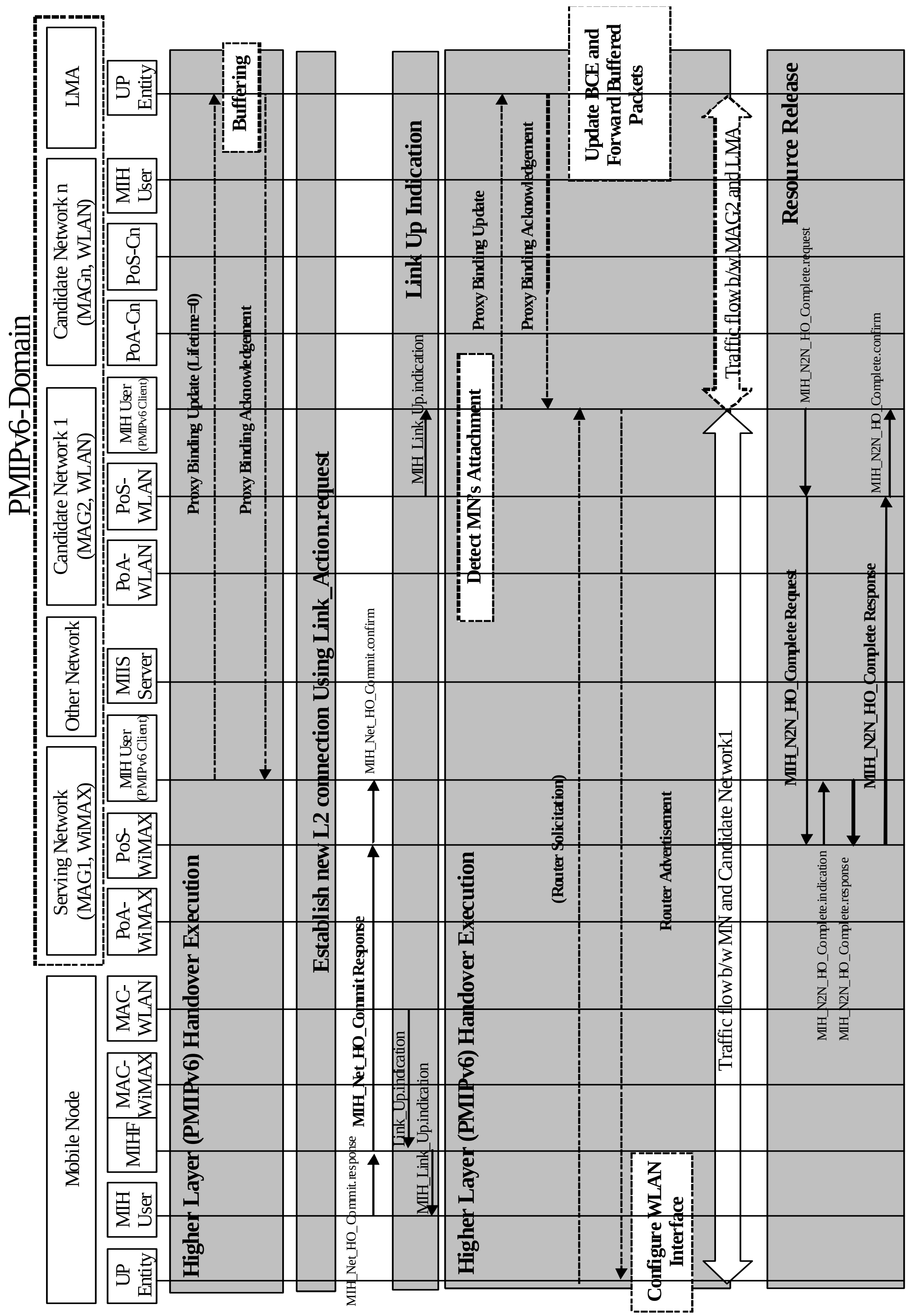


Figure L-4a—Network-initiated handover procedures (cont.)

L.4.2 Mobile-initiated handover procedures

Figure L-5 shows a mobile-initiated handover flow chart for Proxy Mobile IPv6 (PMIPv6), which is currently under standardization for supporting a local mobility in IETF NetLMM Working Group (Although the Proxy Mobile IP is under standardization, its overall flow is already defined. Following handover flow refers to the overall flow). The handover flow operates as follows:

1) MN receives packets through both Mobile Access Gateway (MAG) 1 located in the serving network and Local Mobility Anchor (LMA), which are primary components of the PMIPv6.

2) The MN queries the Information Server to get information about available neighboring networks. This information queries be attempted as soon as the MN attaches to a new serving network or periodically for refreshing the information.

3) MN sends the MIH_MN_HO_Candidate_Query Request message to the Serving PoS for triggering a mobile-initiated handover. This message contains requirements for potential candidate networks.

4) The Serving PoS sends the MIH_N2N_HO_Query_Resource Request messages to the informed Candidate PoSs (can be more than one) in order to query the availability of the resource at the candidate networks. The Candidate PoS responds by sending the MIH_N2N_HO_Query_Resource Response message to the Serving PoS. The Serving PoS in turn sends MIH_MN_HO_Candidate_Query Response message to the MN. Finally, the MN decides the handover target based on the result of query about resource availability at the candidate networks.

5) The MN sends the MIH_MN_HO_Commit Request message to notify the Serving PoS of the decided target network information. The Serving PoS reserves the resource at the target network through MIH_N2N_HO_Commit messages. Upon receiving the MIH_N2N_HO_Commit Request message, PMIPv6 client as MIH User in the target PoS queries the incoming MN's profile to a policy store such as AAA server. As a result, the Target PoS obtains MN's information for PMIP processes in advance. (Upon receiving the MIH_N2N_HO_Commit Request message, PMIPv6 client in the Target PoS queries the incoming MN's profile to an AAA server and send Proxy Binding Update in order to register the location of the MN in advance. The PMIPv6 client in the Target PoS also buffers the packets received from LMA until the MN attaches to the Target PoS.)

6) The Target PoS replies the Serving PoS with the result of the resource preparation by sending MIH_N2N_HO_Commit Response.

7) The MN performs handover to the specified network type and PoA by the MIH_Link_Actions.request primitive. Upon detecting MN's detachment, PMIPv6 client in the Serving PoS terminates a current binding of the MN via sending Proxy Binding Update with Lifetime set to 0 and requests LMA to buffer packets destined for the MN.

8) Once the MN establishes the layer 2 connection to the Target PoS, PMIPv6 client as MIH User in the Target PoS registers the current MN's location to LMA by sending Proxy Binding Update message. LMA updates its Binding Cache Entry with the Proxy Binding Update message and then replies with Proxy Binding Acknowledgement message. LMA also forwards the buffered packets.

9) After receiving the Proxy Binding Acknowledgement message, PMIPv6 clients sends Router Advertisement message to the MN. The Router Advertisement is constructed with the MN's information obtained from the policy server and LMA. It can be solicited by Router Solicitation message from the MN or periodically transmitted. MN configures IP addresses on its interface, which is currently used to connect to the Target PoS, with the received Router Advertisement message. Once the PMIPv6 procedures are completed, MN receives packets through both MAG 2 and LMA.

10) After the PMIPv6 execution, the Target PoS sends the MIH_N2N_HO_Complete Request message to the previous Serving PoS. The previous Serving PoS responds the message with MIH_N2N_HO_Complete Response.

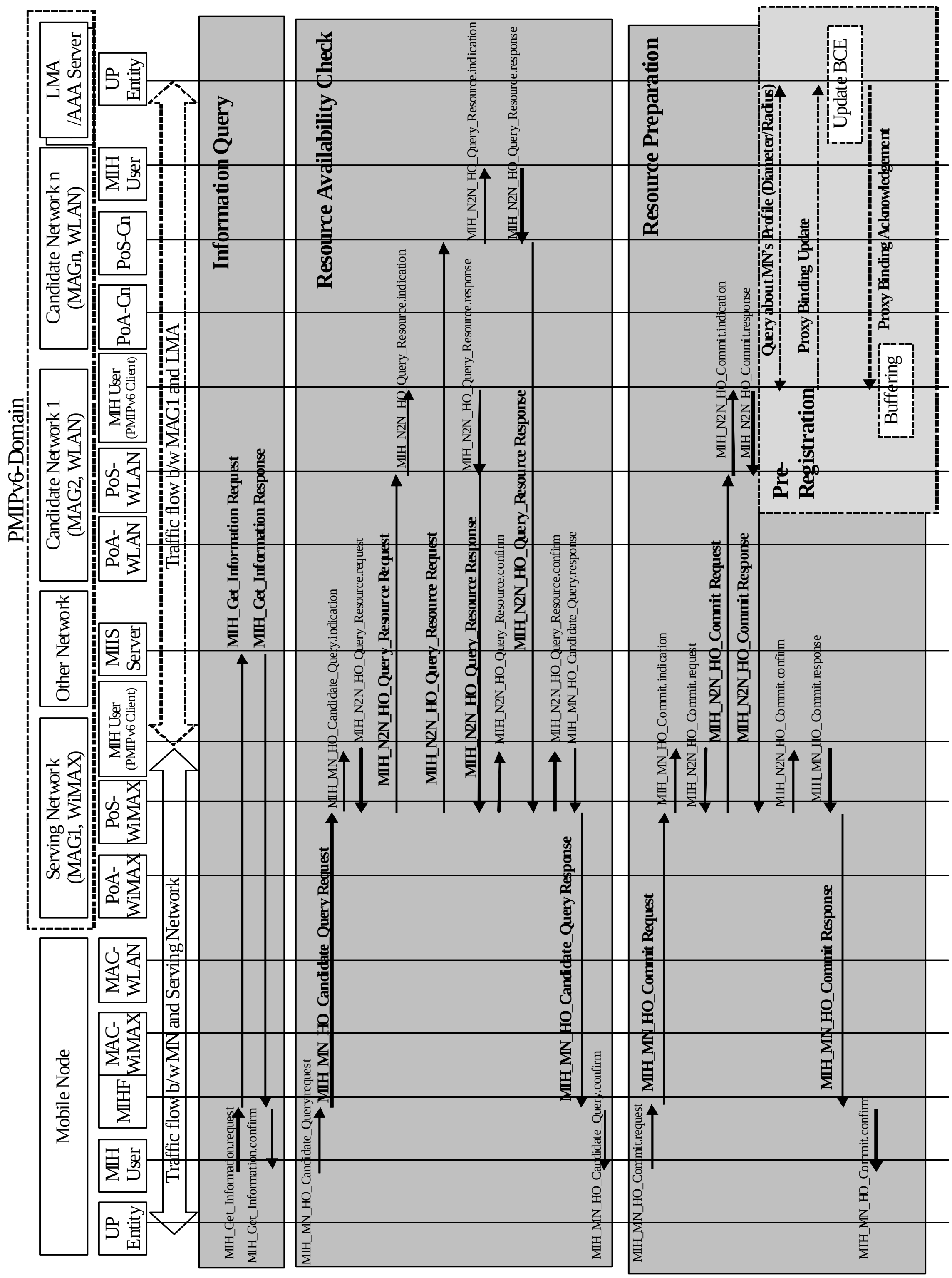


Figure L-5—Mobile-initiated handover procedures

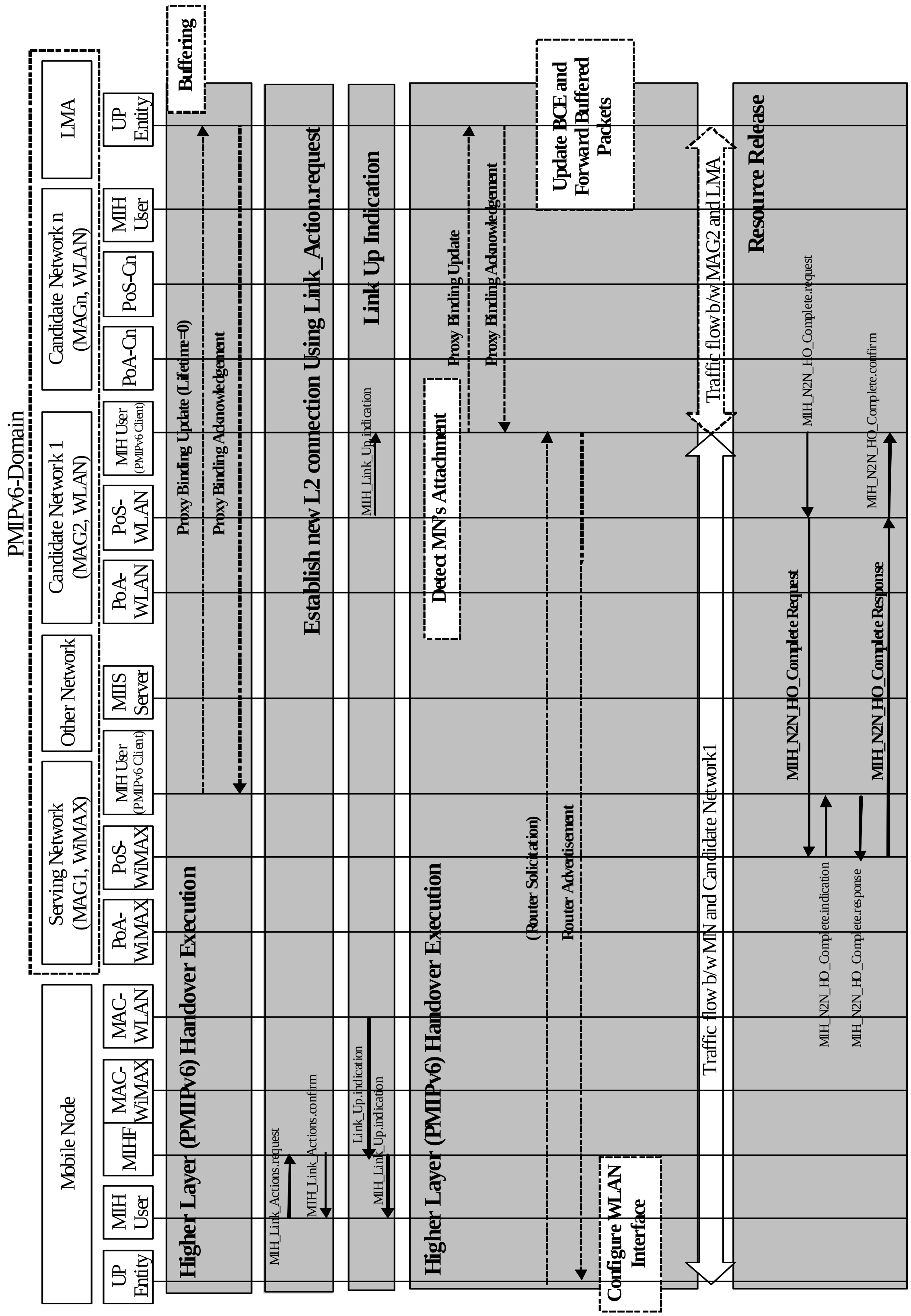
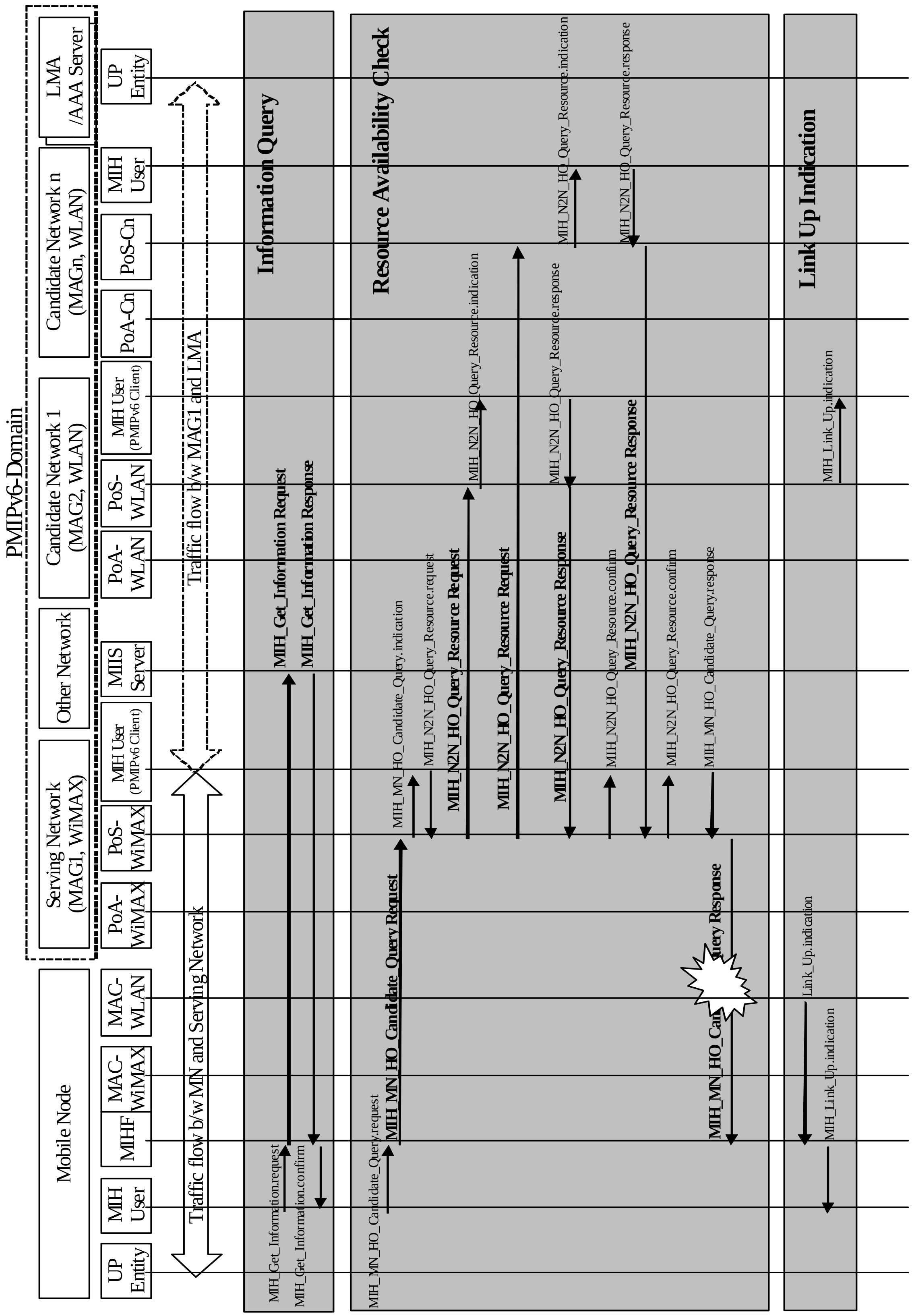


Figure L-5a—Mobile-initiated handover procedures (cont.)



L.4.3 Mobile-initiated handover for break before make case

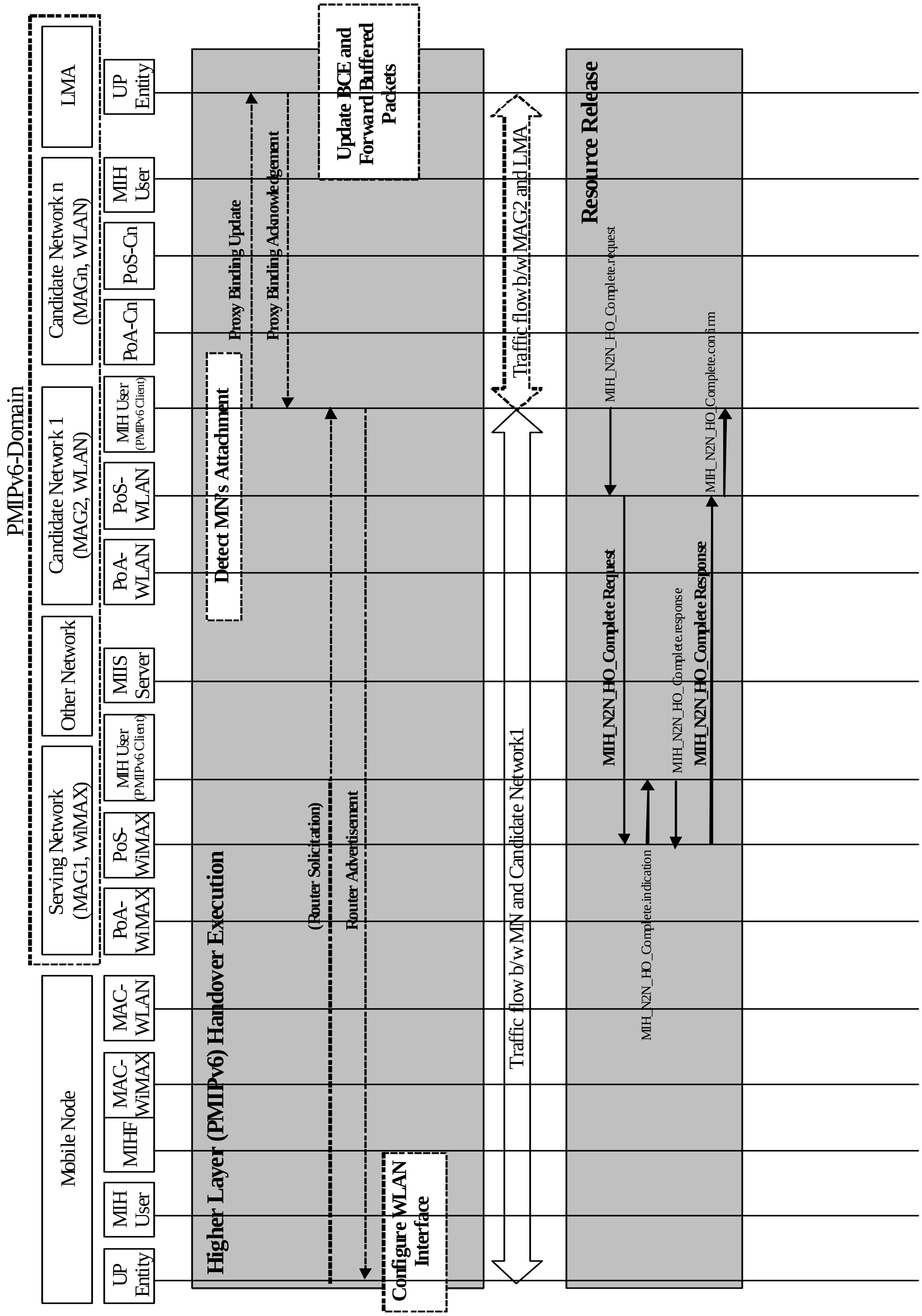


Figure L-6 and Figure L-6a show a mobile-initiated handover flow chart for Proxy Mobile IPv6 (PMIPv6). In this case the MN looses its connectivity with the serving PoA before the target PoA can be notified of MN decision to handover. However, the MN discovers the target PoA, establishes connectivity with the target PoA and then the target PoA notifies serving PoA of handover completion. PMIPv6 signaling is then completed and the packets are then forwarded to MN's new location. The handover flow operates as follows:

1) MN receives packets through both Mobile Access Gateway (MAG) 1 located in the serving network and Local Mobility Anchor (LMA), which are primary components of the PMIPv6.

2) The MN queries the Information Server to get information about available neighboring networks. This information query can be attempted as soon as the MN attaches to a new serving network or periodically for refreshing the information.

3) MN sends the MIH_MN_HO_Candidate_Query Request message to the Serving PoS for triggering a mobile-initiated handover. This message contains requirements for potential candidate networks.

4) The Serving PoS sends the MIH_N2N_HO_Query_Resource Request messages to the informed Candidate PoSs (can be more than one) in order to query the availability of the resource at the candidate networks. The Candidate PoS responds by sending the MIH_N2N_HO_Query_Resource Response message to the Serving PoS. The Serving PoS in turn sends MIH_MN_HO_Candidate_Query Response message to the MN. Finally, the MN decides the handover target based on the result of query about resource availability at the candidate networks.

5) The MN unexpectedly looses connectivity with the serving PoS. Upon detecting MN's detachment, PMIPv6 client in the Serving PoS terminates a current binding of the MN via sending Proxy Binding Update with Lifetime set to 0 and requests LMA to buffer packets destined for the MN.

6) Once the MN establishes the layer 2 connection to the Target PoS, the PMIPv6 client as MIH User in the Target PoS registers the current MN's location to LMA by sending Proxy Binding Update message. LMA updates its Binding Cache Entry with the Proxy Binding Update message and then replies with Proxy Binding Acknowledgement message.

7) After receiving the Proxy Binding Acknowledgement message, PMIPv6 clients sends Router Advertisement message to the MN. The Router Advertisement is constructed with the MN's information obtained from the policy server and LMA. It can be solicited by Router Solicitation message from the MN or periodically transmitted. MN configures IP addresses on its interface, which is currently used to connect to the Target PoS, with the received Router Advertisement message. Once the PMIPv6 procedures are completed, MN receives packets through both MAG 2 and LMA.

10) After the PMIPv6 execution, the Target PoS sends the MIH_N2N_HO_Complete Request message to the previous Serving PoS. The previous Serving PoS responds the message with MIH_N2N_HO_Complete Response.

Figure L-6—Mobile-initiated handover for break before make case

Figure L-6a—Mobile-initiated handover for break before make case (cont.)

L.5 Network selection in 802.11 (WLAN) using 802.21

Figure L-7 shows the general topology of an 802.11 (WLAN) network operating with an 802.21 MIIS.

The steps in network selection as shown in Figure L-8 are as follows:

1) Pre-configuration: The AP is pre-configured with advertising protocol identifier (APID) of choice and is pre-configured to use 802.21 MIIS. The AP discovers the MIIS through a variety of different mechanisms that are outside the scope of specification. The maximum length of response messages from MIIS is also set. The AP communicates with MIIS at L2 or at L3 using a protocol defined elsewhere.

2) Discover AP/Access Network Capabilities: The AP sends out a beacon with Inter-working IE set and APID set to GAS (Generic Advertisement Service). The STA discovers access network capabilities by listening to beacons or it could also send a probe request and discover access network capabilities through the probe response.

3) Query list of subscription service provider networks (SSPNs): The STA sends out a query asking for list of available SSPNs. The query is defined using an 802.21 specific MIH frame. The MIH frame is then relayed by the AP to the MIIS. Meanwhile the AP sends out the initial GAS response to the STA with initial delay (comeback delay).

4) GAS response: The MIIS interprets the query and retrieves the response either from local or remote repository. It then packs the response in an appropriate MIH frame and sends it to the AP. Subsequently when the STA sends the GAS comeback request to the AP, the AP responds with the available information in the MIH frame. The STA then retrieves the information out of the MIH frame and obtains the answer to the query.

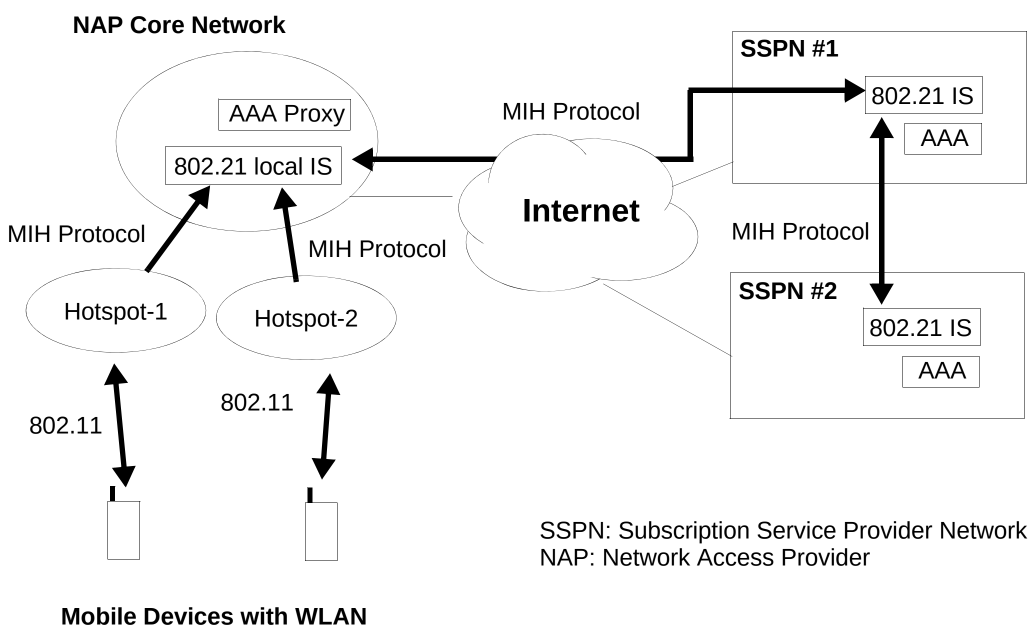


Figure L-7—Network selection in WLAN with 802.11 and 802.21

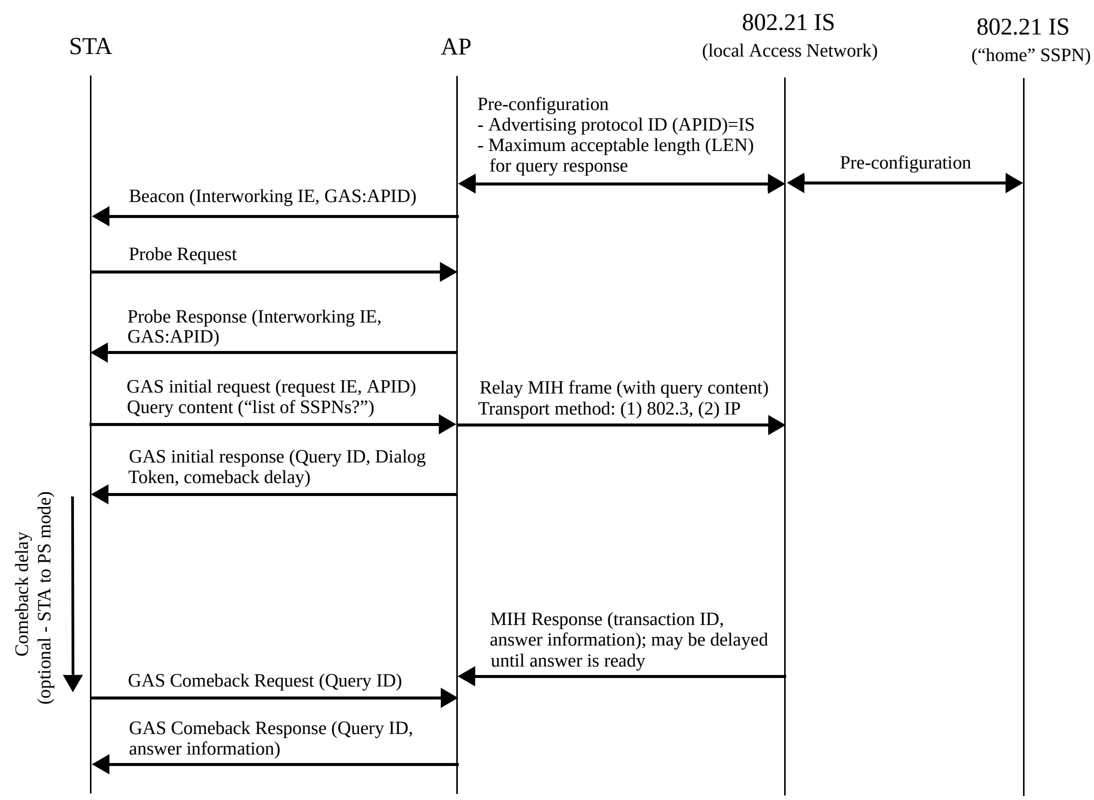


Figure L-8—Use case: query SSPN list

L.6 Handover flow diagram for abnormal cases

L.6.1 Abnormal case at the stage of resource availability check

When MIH_ MN_HO_Candidate_Query Response message indicates the handover initiator to abort handover, the MN will initiate handover process to another suitable target PoA. Figure L-9 shows the handover flow diagram for this case:

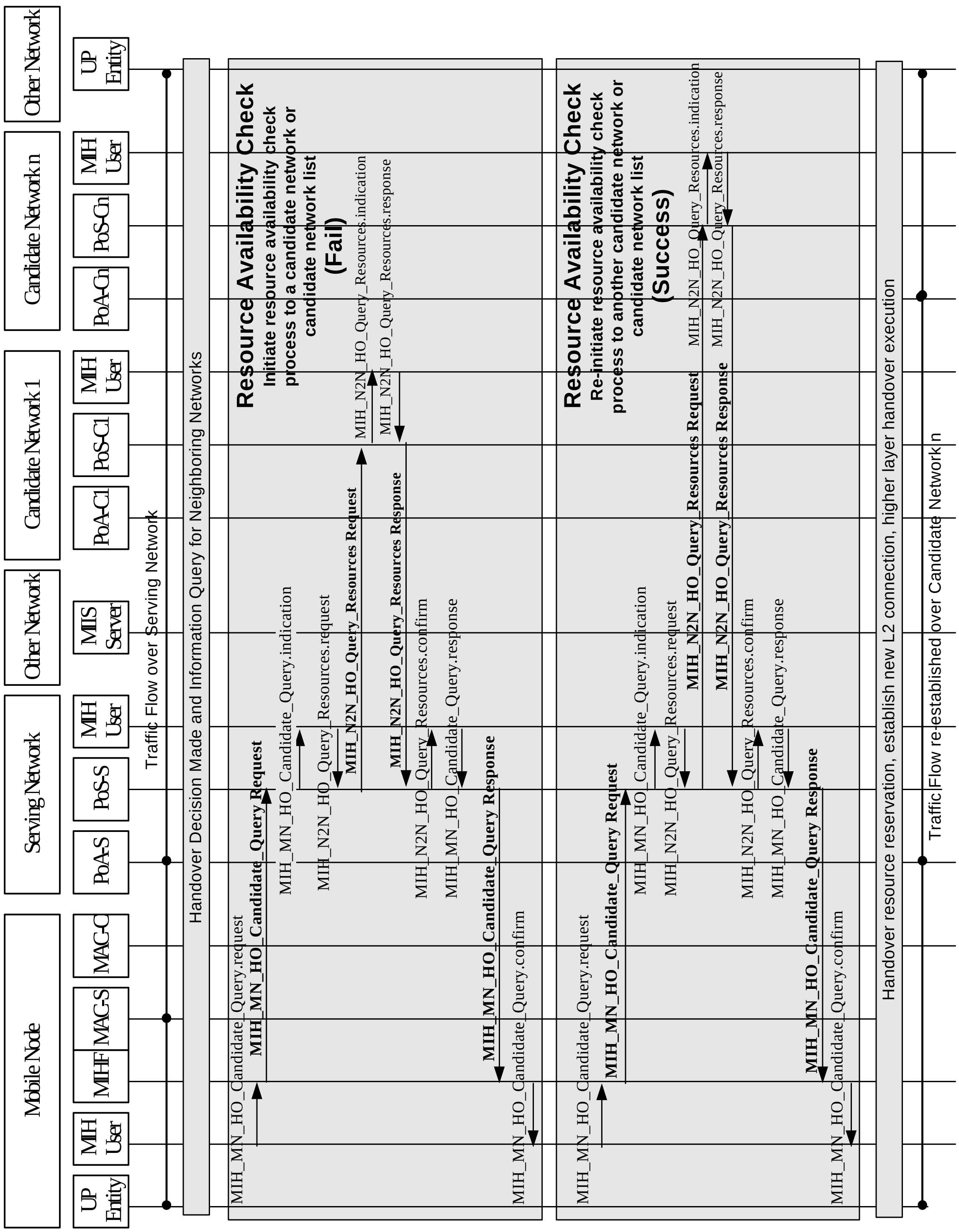


Figure L-9—Abnormal case at the stage of resource availability check

Annex M Example MIH message fragmentation

(informative)

An example of an original MIH message and fragmented MIH messages is shown in Figure M-1.

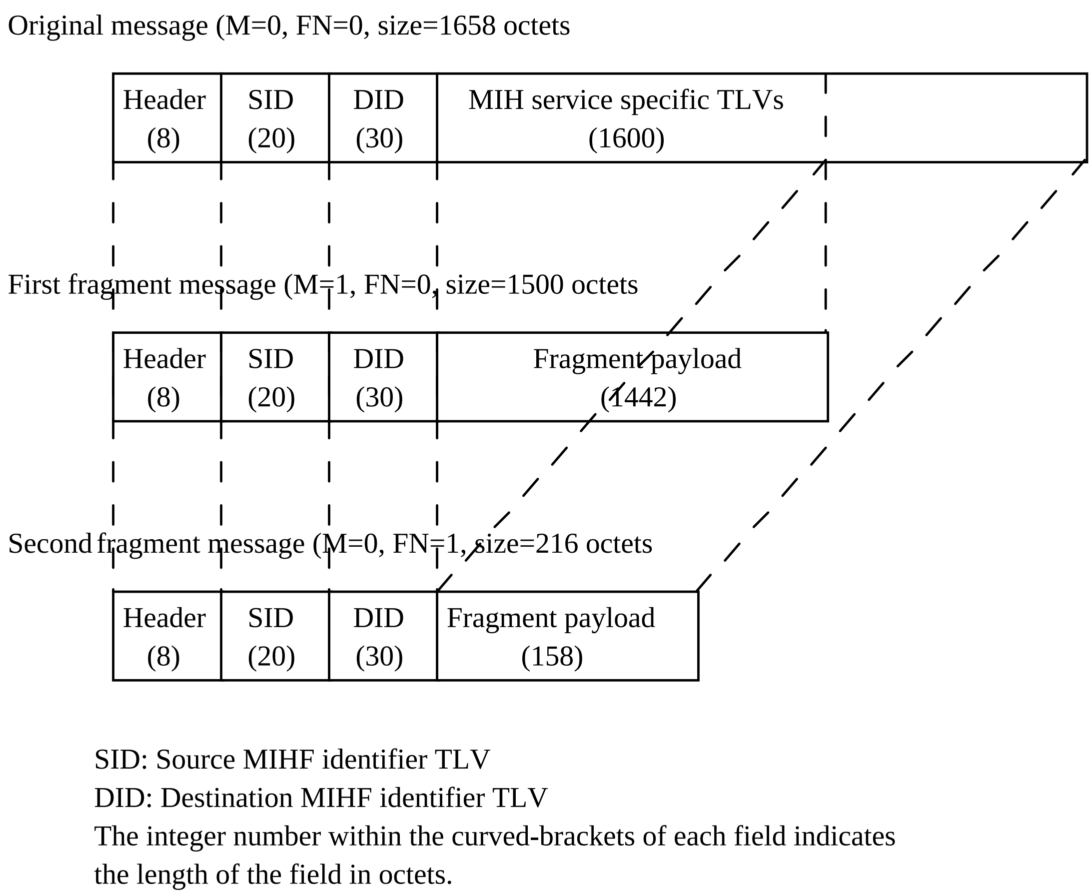


Figure M-1—MIH Fragmentation example for MTU of 1500 octets