IEEE STANDARDS ASSOCIATION

IEEE Standard for a Convergent Digital Home Network for Heterogeneous Technologies

IEEE Communications Society

Sponsored by the
Power Line Communications Standards Committee

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

IEEE Std 1905.1™-2013

12 April 2013

IEEE Standard for a Convergent Digital Home Network for Heterogeneous Technologies

Sponsor

Power Line Communications Standards Committee
of the
IEEE Communications Society

Approved 6 March 2013

IEEE-SA Standards Board

Abstract: An abstraction layer for multiple home networking technologies that provides a common interface to widely deployed home networking technologies is defined in this standard: IEEE 1901 over power lines, IEEE 802.11 for wireless, Ethernet over twisted pair cable, and MoCA 1.1 over coax. Connectivity selection for transmission of packets arriving from any interface or application is supported by the 1905.1 abstraction layer. Modification to the underlying home networking technologies is not required by the 1905.1 layer, and hence it does not change the behavior or implementation of existing home networking technologies. Introduced by the 1905.1 specification is a layer between layers 2 and 3 that abstracts the individual details of each interface, aggregates available bandwidth, and facilitates seamless integration. The 1905.1 also facilitates end−to−end quality of service (QoS) while simplifying the introduction of new devices to the network, establishing secure connections, extending network coverage, and facilitating advanced network management features including discovery, path selection, autoconfiguration, and quality of service (QoS) negotiation.
Keywords: abstraction layer, access point (AP) autoconfiguration, data models, fragmentation and reassembly, IEEE 802.1 bridge discovery, IEEE 802.11™, IEEE 1901™, IEEE 1905.1™, MoCA, pairwise master key, push button, registration, security, topology discovery protocol, wireless fidelity (Wi-Fi)

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

PDF: ISBN 978-0-7381-8297-1 STD98173
Print: ISBN 978-0-7381-8298-8 STDPD98173

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. 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 IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. 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 any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, 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 in its standards 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 ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

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

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

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

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

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854-4141
USA

Photocopies: 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.

Notice to users

Laws and regulations

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

Copyrights

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

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://standards.ieee.org/index.html or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit IEEE-SA Website at http://standards.ieee.org/index.html.

Errata

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

Patents

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

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

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the P1905.1 Working Group had the following membership:

Paul Houzé, *Chair*
Purva Rajkotia, *Vice Chair*
C. Scott Willy, *Editor*

The Working Group gratefully acknowledges the contributions of the following entities and participants. Without their assistance and dedication, this standard would not have been completed.

The following entities submitted technical contributions or commented on the standard at various stages of the project development.

Broadcom
Celeno
Cisco
Cortina Systems
devolo AG
EchoStar
Entropic Communications
Ericsson Inc.
France Telecom
GE
HGI
HomeGrid Forum
HD-PLC Alliance
HomePlug Powerline Alliance
HomePNA Alliance
Ikanos Communications, Inc.
Intel Corporation
Lantiq
Marvell Semiconductor, Inc.
MediaTek
MoCA
MStar
NTT Advanced Technology Corp.
Panasonic Corporation
Qualcomm / Atheros
Ralink
SAGEMCOM SAS
Siemens Corporation
Sigma Designs
Sony Corporation
STMicroelectronics
Toshiba Corporation
Verizon
Vixs Systems, Inc.
ZTE Corporation

The following individuals submitted technical contributions or commented on the standard at various stages of the project development.

Sundeep Ahluwalia
Jim Allen
Avner Aloush
Todd Antes
Mitch Aramaki
Gilles Barberi
Michael Bahr
Frederic Bard
David Barr
Duncan Bees
Ivar Beljaars
Edith Berard
Peter Caldera
William Carney
Regis Cattenoz
Bruce Chang
Joseph Choghi
Philippe Christin
Patrick Clement
Etan Cohen
Olga Cordero-Brana
James Doyle
Jeff Drake
John Egan
Tim Farnham
Jean-Philippe Faure
Norm Finn
Randy Gellens
Navid Ghazisaidi
Jean Grappy
Duncan Ho
Paul Houzé
David Hunter
Aref Iskandar
Rajeev Jain
Gina Jacalne
Georgios Kalogridis
William Keasler
Patrick Keliher
Joon Bae Kim
Neal King
Philippe Klein
Avi Kliger
Michael Koch
Yoshihiro Kondo
Gary Langille
Rick Li
Qiongwen (Jodie) Liang
Oleg Logvinov
Rahul Malik
Jianli Mao
Marcos MartinezEric Masera
Anil Mengi
Cimarron Mittelsteadt
Parag Mogre
Bibha Mohanty
Richard Nesin
Lup Ng
Jordan Nicol
Barry O'Mahony
Txema Ogara
Stephen Palm
Andrea Peccicciono
Martin Renard
Purva Rajkotia
Rob Ranck
Antonio Salas
Roger Samy
Vincenzo Scarpa
Sid Schrum
Andreas Schwager
Kevin Sievert
Nir Shapira
Lydi Smaini
Matt Theall
Rami Verbin
Chao-Chun Wang
James Wang
Lin Wang
C. Scott Willy
Michael Wilson
James Yee
Chiawei Yen
Abdel Younes
Boshan Zhang
Dezhi (James) Zhang
Junjian Zhan

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

Alcatel–Lucent Technologies
Broadcom
Cortina Systems
devolo AG
Duke Energy Corporation
Entropic Communications
FiberHome Technologies Group
France Telecom
Freescale Semiconductor, Inc.
HD-PLC Alliance
HomePlug Powerline Alliance
HomePNA Alliance
Intel Corporation
Marvell Semiconductor, Inc.
Maxim Integrated Products
Mitsubishi Electric Corporation
Motorola Mobility
MoCA
Nokia
NXP Semiconductors
Panasonic Corporation
Power Plus Communications AG
Qualcomm Incorporated
SAGEMCOM SAS
Schneider Electric
Siemens Corporation
Sony Corporation
STMicroelectronics
Toshiba Corporation
Vixs Systems, Inc.
ZTE Corporation

When the IEEE-SA Standards Board approved this standard on 6 March 2013, it had the following membership:

John Kulick, *Chair*
David J. Law, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Peter Balma
Farooq Bari
Ted Burse
Wael William Diab
Stephen Dukes
Jean-Philippe Faure
Alexander Gelman
Mark Halpin
Gary Hoffman
Paul Houzé
Jim Hughes
Michael Janezic
Joseph L. Koepfinger*
Oleg Logvinov
Ron Petersen
Gary Robinson
Jon Walter Rosdahl
Adrian Stephens
Peter Sutherland
Yatin Trivedi
Phil Winston
Yu Yuan

*Member Emeritus

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

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Soo H. Kim
IEEE Client Services Manager, Professional Services

Introduction

This introduction is not part of IEEE Std 1905.1-2013, IEEE Standard for a Convergent Digital Home Network for Heterogeneous Technologies.

Among the home networking technologies, wireless networks offer mobility and wired technologies offer extensive bandwidth or outlet ubiquity for data communications. Wired and wireless technologies complement each other to provide full home coverage.

To address the wide variety of applications, regions, environments, and topologies, multiple connectivity technologies are needed. Each of these different technologies has a unique interface to higher layer entities, thus, leading to software and hardware design complexities in multiconnectivity devices. This complexity must be reduced and new features/functions must be enabled that can take advantage of the multiple paths available between devices.

IEEE Std 1905.1 addresses these requirement by defining an abstraction layer for multiple home networking technologies that provides a common interface to widely deployed home networking technologies: IEEE Std 1901™-2010 over power lines, IEEE Std 802.11™-2012 for wireless, Ethernet over twisted pair cable, and MoCA® 1.1 over coax.[a,b]

[a] Information on references can be found in Clause 2.
[b] MoCA is a registered trademark in the U.S. Patent & Trademark Office, owned by the Multimedia over Coax Alliance.

Contents

Figures

Tables

IEEE Standard for a Convergent Digital Home Network for Heterogeneous Technologies

1. Overview

1.1 Scope

This standard defines an abstraction layer for multiple home network technologies. The abstraction layer provides a common data and control service access point to the heterogeneous home network technologies described in the following specifications: IEEE Std 1901™-2010, IEEE Std 802.11™-2012, IEEE Std 802.3™-2008, and MoCA® 1.1.[1,2] This standard is extensible to work with other home network technologies.

The abstraction layer supports a dynamic interface selection for transmission of packets arriving from any interface (upper protocol layers or underlying network technologies). End-to-end quality of service (QoS) is enabled in an IEEE 1905.1 network.

Also specified are procedures, protocols, and guidelines to provide a simplified user experience to add devices to the network, to set up encryption keys, to extend the network coverage, and to provide network management features to address issues related to neighbor discovery, topology discovery, interface selection, QoS negotiation, and network control and management.

[1] Information on references can be found in Clause 2.
[2] MoCA is a registered trademark in the U.S. Patent & Trademark Office, owned by the Multimedia over Coax Alliance.

1.2 Purpose

The abstraction layer's common interface allows applications and upper layer protocols to be agnostic to the underlying network technologies. The purpose of this standard is to facilitate the integration of IEEE 1901 with other home network technologies.

Additionally, the purpose of this standard is to define an abstraction layer that allows the following:

- Common network setup among heterogeneous network technologies
- Providing the same user experience in the process of adding a device to the network and the same user experience while setting an encryption key
- Intelligent network interface selection for delivery of packets that provides improved coverage performance, improved data rate on the poorest link, improved network capacity, improved network reliability and QoS, and support for end-to-end QoS for different traffic classes
- Seamless/transparent interface switching
- Real-time mapping of connection links and interfaces for each traffic class/stream
- Green energy management

2. Normative references

The following referenced document is indispensable for the application of this document (i.e., it 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.

Connection Handover Technical Specification NFC Forum, Connection Handover 1.2; NFCForum-TS-ConnectionHandover_1_2.doc; 2010-07-07.[3]

FIPS PUB 180-2:2002, Secure Hash Signature Standard (SHS) (including the change notice dated February 25, 2004, concerning truncation).[4]

IEEE P802.11ac/D3.0 (June 2012), Enhancements for Very High Throughput for Operation in Bands below 6 GHz.[5]

IEEE Std 1901™-2010, IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications.[6,7]

IEEE Std 802™-2001, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.

IEEE Std 802.1AB™-2009, IEEE Standard for Local and Metropolitan Area Networks—Station and Media Access Control Connectivity Discovery.

IEEE Std 802.1D™-2004, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges.

[3] NFC Forum publications are available from the NFC Forum Technical Specifications (http://www.nfc-forum.org/).
[4] FIPS publications are available from the National Technical Information Service (http://www.ntis.gov/).
[5] This IEEE standards project was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.
[6] IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).
[7] The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

IEEE Std 802.1Q™-2011, IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks.

IEEE Std 802.3™-2008, IEEE Standard for Information Technology-Specific Requirements—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

IEEE Std 802.11™-2012, IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**.**

IEEE Std 802.11ad™-2012, IEEE Standard for Information technology--Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band.

IETF RFC-6234, U.S. Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF).[8]

MoCA® MAC/PHY Specification v1.1, MoCA-M/P-SPEC-V1.1-06272011, Multimedia over Coax Alliance (MoCA).[9]

PKCS #5 V2.0-1999, Password-Based Cryptography Specification.[10]

Simple Object Access Protocol (SOAP) 1.1.[11]

TR-069 Amendment 4, July 2011, CPE WAN Management Protocol, Broadband Forum.[12]

TR-106 Amendment 6, July 2011, Data Model Template for TR-069-Enabled Devices, Broadband Forum.

TR-181 Issue: 2, May 2010, Device Data Model for TR-069, Broadband Forum Technical Report.

US-ASCII, ANSI_X3.4-1968, Coded Character Sets—7-Bit American National Code for Information Exchange.[13]

Wi-Fi® Simple Configuration (WSC), Wi-Fi Simple Configuration Technical Specification Version 2.0.2.[14]

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[15, 16]

[8] IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.
[9] MoCA specifications are available from the Multimedia over Coax Alliance (http://www.mocalliance.org/specs).
[10] RSA Laboratories specifications are available from RSA Laboratories (http://www.rsa.com/). This document is available for download at ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf/.
[11] This document is available for download at http://www.w3.org/TR/2000/NOTE-SOAP-20000508.
[12] Broadband Forum Technical Reports are available from the Broadband Forum (http://www.broadband-forum.org/).
[13] ANSI publications are available from the American National Standards Institute (http://www.ansi.org/).
[14] World Wide Web Consortium (W3C) publications are available from the World Wide Web Consortium (http://www.w3c.org/). Wi-Fi is a registered trademark in the U.S. Patent & Trademark Office, owned by the Wi-Fi Alliance.
[15] The *IEEE Standards Dictionary Online* subscription is available at http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

1905.1 abstraction layer: A layer between the logical link control (LLC) and one or multiple media access control (MAC) service access points (SAPs) of 1905.1 supported MAC/physical layer (PHY) standards. The 1905.1 abstraction layer (AL) is identified by its 1905.1 AL MAC address.

1905.1 abstraction layer management entity (ALME): A management entity that provides a management service interface to the 1905.1 abstraction layer and the underlying network technologies.

1905.1 abstraction layer (AL) MAC address: A locally administered Extended Unique Identifier-48 (EUI-48) value that uniquely identifies a 1905.1 abstraction layer.

1905.1 management entity: An entity responsible for generating and processing 1905.1 interabstraction layer messages.

1905.1 device: A device with one or more interfaces abstracted by a 1905.1 abstraction layer.

1905.1 higher-layer entity (HLE): A higher layer entity that has the ability to make use of a 1905.1 abstraction layer (AL).

1905.1 interface: An underlying network interface [media access control/physical layer (MAC/PHY)] technology that is supported by IEEE Std 1905.1.

1905.1 link: A logical link set by a 1905.1 management entity between the 1905.1 abstraction layers of two particular 1905.1 devices through their respective 1905.1 interfaces to exchange 1905.1 control messages (CMDUs) and datagrams [protocol data units (PDUs)].

1905.1 link metrics: The link metrics of a transmission channel of a 1905.1 interface.

1905.1 multicast media access control (MAC) address: The destination address of messages using the registered group MAC address: 01-80-C2-00-00-13.

1905.1 network: A set of 1905.1 devices interconnected by 1905.1 links.

1905.1 network key: The privacy parameter shared by all 1905.1 devices from which u-key(s) are derived.

access point (AP) enrollee: An AP that accepts IEEE 802.11 parameters (e.g., security credentials) from a registrar based on the IEEE 1905.1 AP autoconfiguation protocol.

authenticated 1905.1 interface: A 1905.1 interface that transfers media access control (MAC) service data units (MSDUs) over an encrypted link if the underlying network technology encryption mode is enabled. When the underlying network technology does not support encryption mode or the encryption of the underlying MAC/physical layer (PHY) is disabled for this link, the 1905.1 interface is considered authenticated through NULL authentication.

bridging tuple: A list of media access control (MAC) addresses of a 1905.1 device's network interfaces for which packets can be forwarded between the interfaces.

discovery: A set of procedures allowing a 1905.1 higher layer entity (HLE) to discover other 1905.1 devices and detect the presence of IEEE 802.1 bridges between the HLE's 1905.1 device and another 1905.1 device.

higher layer entity (HLE): An entity above the 1905.1 abstraction layer management entity (ALME) interface.

IEEE 802.1 bridge: A bridge that complies with IEEE Std 802.1D™-2004 or IEEE Std 802.1Q but does not comply with IEEE Std 1905.1.

lost packet: A media access control (MAC) protocol data unit (MPDU) that arrives at the link transmitter but is never received at the intended link receiver or is received with cyclic redundancy check (CRC) errors.

media access control (MAC) address: Extended Unique Identifier-48 (EUI-48) MAC address as described in IEEE Std 802®-2001.

neighbor 1905.1 device: A 1905.1 device connected by at least one 1905.1 link.

[16] Key IEEE 1905.1 concepts are referred to from here on as either "IEEE 1905.1" or just "1905.1."

neighbor multicast control message data unit (CMDU): A 1905.1 multicast message that is not retransmitted by a receiving 1905.1 device.

non-1905 neighbor device: A device that originates traffic, connected by at least one link that is not a 1905.1 device.

notification: Procedures by which a 1905.1 management entity notifies other 1905.1 control entities that a change in the sender's topology has occurred.

push button event: An event triggered by pressing a physical or logical button to authenticate an 1905.1 interface through a push button configuration method. The user experience information is defined in H.5 of IEEE Std 1901-2010.

registrar: An entity that issues IEEE 802.11 parameters (e.g., security credentials) to a 1905.1 access point (AP) enrollee based on the IEEE 1905.1 AP autoconfiguration protocol.

relayed multicast control message data unit (CMDU): A 1905.1 multicast message that is retransmitted on all the interfaces of the receiving device (except for the interface on which the message was received).

reserved (bits): Bit fields reserved for future revisions of this specification.

reserved values: Values reserved for future revisions of this specification.

station management entity (SME): A management entity that provides a station's media access control (MAC) and physical layer (PHY) management service interface to the abstraction layer management entity (ALME) and higher layer entities (HLEs).

u-key: A generic term to refer to the privacy parameter of each underlying 1905.1 network technology: Wi-Fi wireless protected access (WPA)/wireless protected access II (WPA2) passphrase, IEEE 1901 shared key device-based security network (DSNA) network membership key (NMK), IEEE 1901 pairwise security network (PSNA) pairwise key (PWK), and MoCA privacy password.

unconfigured IEEE 802.11 access point (AP): A 1905.1 device containing a 1905.1 interface, which is an IEEE 802.11 AP station but without any initial configuration parameters [e.g., service set identification (SSID), channel, authentication, and key].

3.2 Acronyms and abbreviations

AL	abstraction layer
ALME	abstraction layer management entity
AP	access point
CMDU	control message data unit
DSNA	device-based security network
KCD	key carrying device
HLE	higher layer entity
L2	layer 2
LLC	logical link control
LLCDU	logical link control data unit
LLDP	link layer discovery protocol
LLDPDU	LLDP data unit
MAC	media access control
MPDU	MAC protocol data unit
MSDU	MAC service data unit
NDEF	near-field communication data exchange format
NFC	near-field communication
NFCNK	near-field communication network key
NMK	network membership key
PDU	protocol data unit
PHY	physical layer
PSNA	pairwise security network

SAP	service access point
SME	station management entity
SSID	service set identifier
TLV	type length value
WPA®	wireless protected access[17]
WPA2	wireless protected access II
WSC	Wi-Fi simple configuration

4. General description

4.1 Introduction

With the increase of bandwidth, intensive home networking applications, and consumers' endless appetite for services, home network technologies have become the latest frontier in the evolution of service delivery.

Both wired and wireless home network technologies have significant market presence due to the value they create for end users. Wireless networks offer mobility, whereas wired technologies offer extensive bandwidth or outlet ubiquity. Wired and wireless technologies complement each other to provide full home coverage.

To address the wide variety of applications, regions, environments, and topologies, multiple connectivity technologies are needed. Over the last 10 years, more than 1 billion home networking devices have been deployed in the market, and hence, any proposed solution must interoperate with this deployed base.

4.2 IEEE Std 1905.1 overview

IEEE Std 1905.1 defines an abstraction layer that provides a common interface to several home network technologies: IEEE 1901 over power lines, Wi-Fi/IEEE 802.11 for wireless, Ethernet over twisted pair cable, and MoCA 1.1 over coax (see Figure 4-1).

The 1905.1 abstraction layer supports connectivity selection for transmission of packets arriving from any interface or application.

The 1905.1 layer does not require modification to the underlying network technologies and hence does not change the behavior or implementation of existing home network technologies.

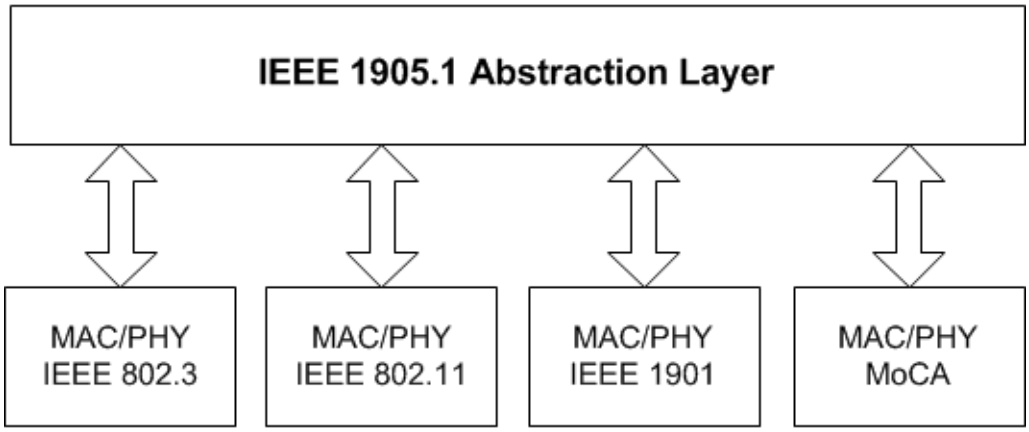

Figure 4-1—Overview of IEEE Std 1905.1

[17] WPA is a registered trademark in the U.S. Patent & Trademark Office, owned by the Wi-Fi Alliance.

The goal of IEEE Std 1905.1 is to define a common fabric that spans established home network technologies and to define a common data and control service access point. Packets can arrive and be transmitted over any of the defined interfaces, regardless of the upper protocol layers or underlying network technologies.

Specifically, IEEE Std 1905.1 introduces an intermediate sublayer between the logical link control (LLC) layer two (L2) sublayer and underlying network technology MAC sublayer(s) that abstracts the individual details of each interface, aggregates available bandwidth, and facilitates seamless integration. This layer simplifies setup, for example, by eliminating the need for a user to enter different passwords to access each link. IEEE Std 1905.1 also enables end-to-end QoS while simplifying the introduction of new devices to the network, establishing secure connections, extending network coverage, and providing advanced network management features including discovery and interface selection.

4.2.1 Benefits of IEEE Std 1905.1

The benefits of IEEE Std 1905.1 include the following:

- Ease of use: It is imperative that network setup and use is simple to consumers. IEEE Std 1905.1 provides common setup procedures for adding devices to a network, establishing secure links, implementing QoS, and managing the network.
- Fallback: When a link goes down temporarily or is congested, an alternative route is available. This reduces the number of problems and interruptions that users experience, as well as reduces the number of support calls that service providers must manage.
- Aggregated throughput: The ability to use a hybrid network's entire available throughput across the different interface networks maximizes throughput.
- Multiple simultaneous streams: IEEE Std 1905.1 helps ensure that multiple simultaneous streams operate smoothly.
- Load balancing: The network can balance bandwidth usage over different interfaces to limit congestion and maintain reliability.
- QoS: IEEE Std 1905.1 enables end-to-end QoS without compromising the internal integrity of any protocol or medium over which it runs.
- Backward compatibility: IEEE Std 1905.1 helps ensure backward compatibility with previous deployments of IEEE 1901, Wi-Fi/IEEE 802.11, Ethernet, and MoCA. IEEE Std 1905.1 provides interoperability with deployed technologies.
- Security: IEEE Std 1905.1 enables consistent password and authentication procedures for non-1905.1 devices.
- Advanced diagnostics: IEEE Std 1905.1 allows the overall network to monitor itself to help ensure reliable and uninterrupted operation.
- Self-install: IEEE Std 1905.1 allows devices to be configured the same way with a simple button push. Pairing is kept simple through a standard push button or NFC mechanisms, thus, avoiding complex configuration. Manual configuration is an option.
- Universal connectivity: IEEE Std 1905.1 allows users to connect to the hybrid network (a network composed of more than one subnetwork types that, when interconnected together, operate in a manner equivalent to that of a network composed of only one subnetwork type) from any room in a house without having to be aware of which part of the network their device is currently interfacing with. Supports handover from one network interface to another as needed when moving from room to room.
- Energy management: Optimizing network power usage across different technologies results in more energy-efficient operations.

4.3 IEEE 1905.1 architecture

The 1905.1 abstraction layer is an intermediate layer between the LLC L2 layer and underlying MAC layer(s) as illustrated in Figure 4-2.

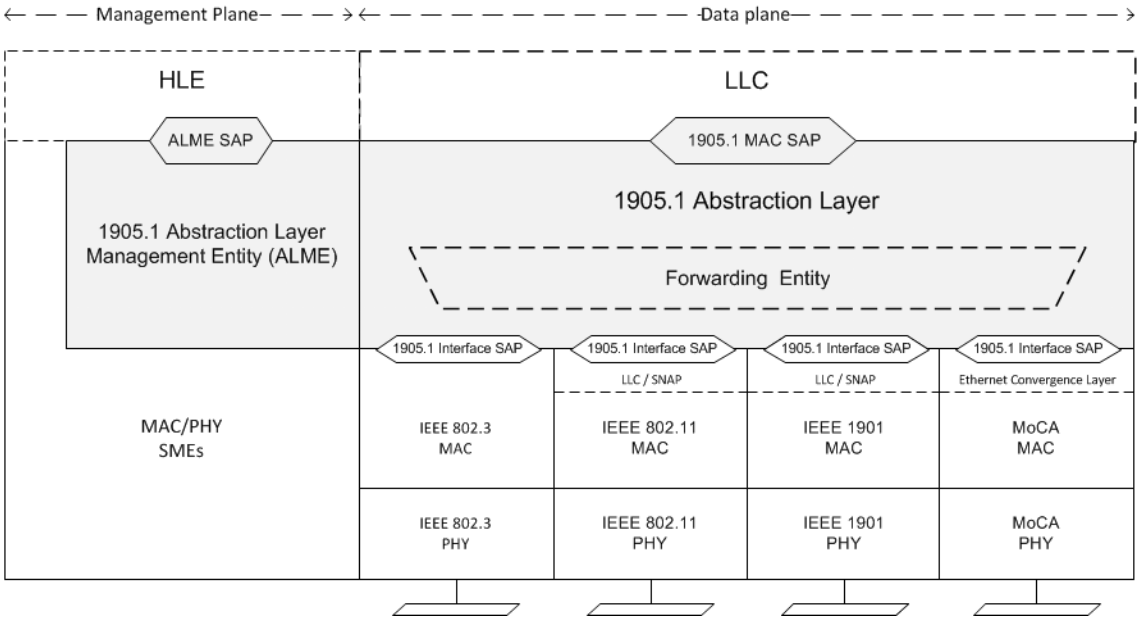

Figure 4-2—1905.1 abstraction layer model

The 1905.1 abstraction layer abstracts the heterogeneous MAC and PHY technologies of the converged home network by creating a single virtual MAC on top of the underlying MAC/PHY of the respective network technologies.

The 1905.1 abstraction layer provides service access points (SAPs) toward the upper layers as follows:

- For the data plane, a 1905.1 MAC SAP with the LLC
- For the management plane, a 1905.1 ALME CTRL SAP to invoke the abstraction layer management functions

The 1905.1 abstraction layer can forward 802.3 MPDUs between the following:

- The 1905.1 MAC SAP and the underlying 1905.1 interfaces
- The underlying 1905.1 interfaces

The behavior of the 1905.1 forwarding entity is not defined in this standard. The forwarding entity, if present, shall be interoperable with IEEE 802.1 bridging. All clauses that describe data forwarding and forwarding rule management in this standard are only applicable if the 1905.1 forwarding entity is present.

A 1905.1 network is formed of 1905.1 devices interconnected through 1905.1 links as illustrated by Figure 4-3.

CMDUs are exchanged between 1905.1 abstraction layers. All CMDUs are received by neighboring 1905.1 abstraction layers. Some types of received CMDU are relayed to other 1905.1 abstraction layers.

Ethernet logical link control data units (LLCDUs) are exchanged between 1905.1 abstraction layers. If an optional 1905.1 forwarding entity is present, then based on the forwarding rules in the 1905.1 abstraction

layer forwarding entity, the Ethernet LLCDUs are forwarded to the LLC and/or forwarded to other 1905.1 abstraction layers.

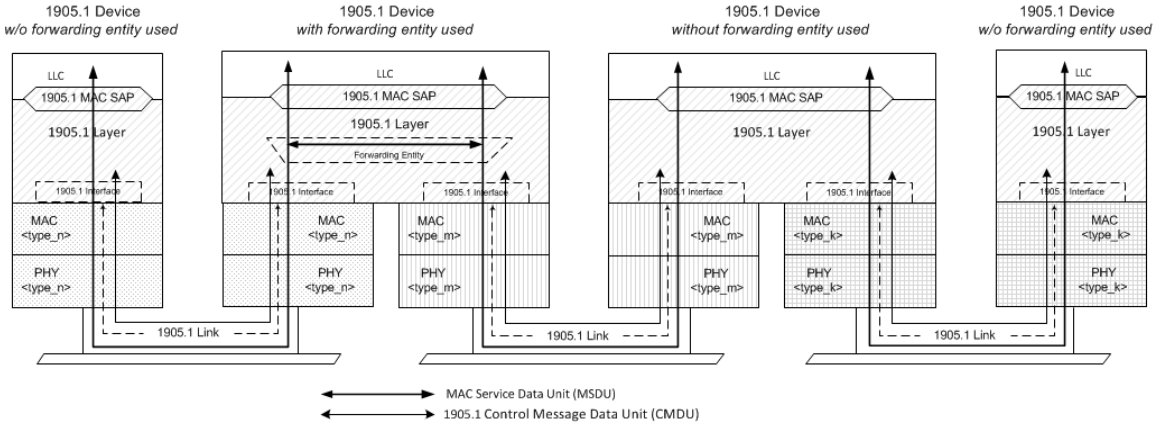

Figure 4-3—1905.1 network

4.4 Abstraction layer

The 1905.1 abstraction layer within a 1905.1 device uses an EUI-48 value (1905.1 AL MAC address) for identification. This 1905.1 AL MAC address may be used as a source or destination address for data and CMDUs originating or destined for the 1905.1 device, respectively.

Each 1905.1 AL shall locally administer its 1905.1 AL MAC address (see definition in 3.1) so that it does not conflict with any other MAC address or 1905.1 AL MAC address in the 1905.1 network to which it connects.

5. IEEE 1905.1 abstraction layer management entity

The 1905.1 ALME provides the layer management service interface through which layer management functions can be invoked to (1) the AL and (2) the underlying network technology SMEs or specific management interfaces that are out of the scope of this standard.

The ALME typically performs such functions on behalf of the HLEs and may implement management protocols that are out of the scope of this standard. Figure 5-1 depicts the relationship among the management entities.

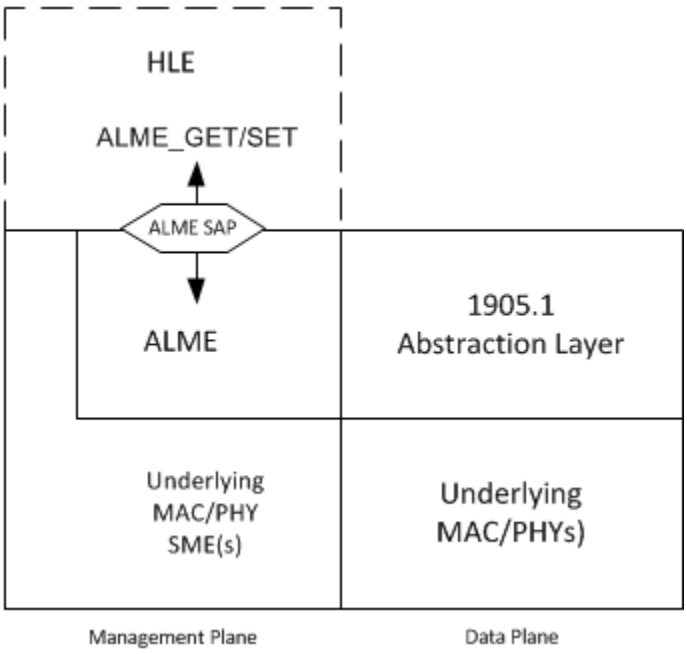

Figure 5-1—Abstraction layer management model

The ALME SET primitives are represented as ".request" with associated ".confirm" primitives, and the ALME GET primitives are represented as ".request" with associated ".response" primitives.

5.1 AL management specific service (ALME-SAP)

5.1.1 ALME-GET-INTF-LIST.request

5.1.1.1 Function

This primitive is used by the HLEs to get a description of the 1905.1 interfaces of the 1905.1 AL for the HLE's device.

5.1.1.2 Semantics of the service primitive

The primitive parameters are as follows:

ALME-GET-INTF-LIST.request (
)

5.1.1.3 When generated

This primitive is generated by an HLE to get the description of the 1905.1 interfaces.

5.1.1.4 Effect of receipt

If the ALME receives the ALME-GET-INTF-LIST.request primitive, then the ALME shall generate an ALME-GET-INTF-LIST.response primitive.

5.1.2 ALME-GET-INTF-LIST.response

5.1.2.1 Function

This primitive is used by the ALME to send a response to the request to get the list of 1905.1 interfaces of the 1905.1 device.

5.1.2.2 Semantics of the service primitive

The primitive parameters are as follows:

ALME-GET-INTF-LIST.response (
intfList
)

Table 5-1 details the parameters for the ALME-GET-INTF-LIST.response.

Table 5-1—ALME-GET-INTF-LIST.response parameters

Name	Type	Valid range	Description
intfList	A set of intfDescriptors	As defined in Table 5-2	The parameters associated with the list of 1905.1 interfaces of the device

Each intfDescriptor consists of the elements detailed in Table 5-2.

Table 5-2—intfDescriptor elements

Name	Type	Valid range	Description
intfAddress	EUI-48 Address	Any MAC address	The physical MAC address of the underlying network technology MAC
intfType	Enumeration	As defined in the first column of Table 6-12	Indicates the MAC/PHY type of the underlying network technology
IEEE802.1BridgeFlag	Boolean	TRUE,FALSE	Boolean flag set as described in 8.1 to indicate that the 1905.1 neighbor device is connected on this particular interface: — Through one or more IEEE 802.1 bridges (TRUE) — Otherwise (FALSE)
vendorSpecificInfo	A set of information elements	As defined in Table 5-20	Zero or more information elements

5.1.2.3 When generated

This primitive is generated by the 1905.1 ALME as a response to an ALME-GET-INTF-LIST.request.

5.1.2.4 Effect of receipt

The intfList provides to the HLEs that generated the ALME-GET-INTF-LIST.request a list of descriptors of individual MAC/PHY abstracted by the 1905.1 AL. This list allows the HLE through the ALME to

manage each MAC/PHY directly either through its standard SME primitives or through implementation-specific functions.

5.1.3 ALME-SET-INTF-PWR-STATE.request

5.1.3.1 Function

This primitive is used to transition the power state of a 1905.1 interface.

5.1.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
ALME-SET-INTF-PWR-STATE.request (
                                    intfAddress,
                                    powerState
                                )
```

Table 5-3 details the parameters for the ALME-SET-INTF-PWR-STATE.request.

Table 5-3—ALME-SET-INTF-PWR-STATE.request parameters

Name	Type	Valid range	Description
intfAddress	EUI-48 Address	Any MAC address	The physical MAC address of the underlying network technology MAC.
powerState	Enumeration	PWR_ON, PWR_SAVE, PWR_OFF	1905.1 interface's power state as defined in Table 5-4.

Table 5-4—Power state to power state field

powerState	powerState field value	Meaning
PWR_ON	0x00	Interface is providing full functionality and full performance for propagating PDUs.
PWR_SAVE	0x01	Interface is in an underlying network technology-specific power-save mode providing limited functionality or limited performance that may incur delay for propagating PDUs.
PWR_OFF	0x02	Interface is unable to propagate PDUs.
Reserved Values	0x03 ~ 0xFF	

5.1.3.3 When generated

This primitive is generated by an HLE to request a 1905.1 interface to transition to a specified power state.

5.1.3.4 Effect of receipt

This request sets the power state of the 1905.1 interface. The receiving ALME subsequently issues an ALME-SET-INTF-PWR-STATE.confirm primitive that reflects the results of the power state set request.

If the ALME receives an ALME-SET-INTF-PWR-STATE.request with a powerState supported by the specified 1905.1 interface, then the 1905.1 interface should transition to the requested powerState and the ALME response in the resulting ALME-SET-INTF-PWR-STATE.confirm shall contain a reasonCode set to the value SUCCESS.

If the ALME receives an ALME-SET-INTF-PWR-STATE.request with a powerState unsupported by the specified 1905.1 interface, then the ALME response in the resulting ALME-SET-INTF-PWR-STATE.confirm shall contain a reasonCode set to the value UNSUPPORTED_PWR_STATE.

If the ALME receives an ALME-SET-INTF-PWR-STATE.request with a MAC address that does not match any 1905.1 interface MAC address, then the ALME response in the resulting ALME-SET-INTF-PWR-STATE.confirm shall contain a reasonCode set to the value UNMATCHED_MAC_ADDRESS.

If the power state is supported but currently unavailable, then the ALME response in the resulting ALME SET-INTF-PWR-STATE.confirm shall contain a reasonCode set to the value UNAVAILABLE_ PWR_STATE.

5.1.4 ALME-SET-INTF-PWR-STATE.confirm

5.1.4.1 Function

This primitive confirms the completion of the power state transition requested by an ALME-SET-INTF-PWR-STATE.request.

5.1.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
ALME-SET-INTF-PWR-STATE.confirm (
                                    intfAddress,
                                    reasonCode
                                )
```

Table 5-5 details the parameters for the ALME-SET-INTF-PWR-STATE.confirm.

Table 5-5—ALME-SET-INTF-PWR-STATE.confirm parameters

Name	Type	Valid range	Description
intfAddress	EUI-48 address	Any MAC address	The intfAddress provided in the ALME-SET-INTF-PWR-STATE.request
reasonCode	Enumeration	SUCCESS, UNAVAILABLE_PWR_STATE, UNMATCHED_MAC_ADDRESS, UNSUPPORTED_PWR_STATE	See Table 5-19

5.1.4.3 When generated

If the requested power state is supported by the 1905.1 interface, then this primitive is generated by the ALME as a response to an ALME-SET-INTF-PWR-STATE.request after the 1905.1 interface transitions to the requested power state.

If the power state transition requested by an ALME-SET-INTF-PWR-STATE.request was completed successfully, then the ALME response in the resulting ALME-SET-INTF-PWR-STATE.confirm shall contain a reasonCode set to the value SUCCESS. If the ALME-SET-INTF-PWR-STATE.request has a MAC address that does not match any 1905.1 interface MAC address, then the reasonCode shall be set to the value UNMATCHED_MAC_ADDRESS.

If the power state is supported but currently unavailable, then the ALME response in the resulting ALME-SET-INTF-PWR-STATE.confirm shall contain a reasonCode set to the value UNAVAILABLE_ PWR_STATE.

If the power state is unsupported, then the ALME response in the resulting ALME-SET-INTF-PWR-STATE.confirm shall contain a reasonCode set to the value UNSUPPORTED_PWR_STATE.

5.1.4.4 Effect of receipt

The requesting HLE is informed of the result of its ALME-SET-INTF-PWR-STATE.request.

5.1.5 ALME-GET-INTF-PWR-STATE.request

5.1.5.1 Function

This primitive is used by HLEs to retrieve the current power state of a 1905.1 interface.

5.1.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
ALME-GET-INTF-PWR-STATE.request (
                                    intfAddress,
                                )
```

Table 5-6 details the parameters for the ALME-GET-INTF-PWR-STATE.request.

Table 5-6—ALME-GET-INTF-PWR-STATE.request parameters

Name	Type	Valid range	Description
intfAddress	EUI-48 address	Any MAC address	The physical MAC address of the underlying network technology MAC

5.1.5.3 When generated

This primitive is generated by HLEs to retrieve the power state of a 1905.1 interface.

5.1.5.4 Effect of receipt

This request retrieves the power state of the specified 1905.1 interface. The receiving ALME subsequently issues an ALME-SET-INTF-PWR-STATE.response primitive that reflects the results of the power state get request.

If the ALME receives an ALME-GET-INTF-PWR-STATE.request with a MAC address that does not match any of the device's 1905.1 interface MAC addresses, then the ALME response in the resulting ALME-GET-INTF-PWR-STATE.response shall contain a reasonCode set to the value UNMATCHED_MAC_ADDRESS. If the ALME receives an ALME-GET-INTF-PWR-STATE.request and issues a subsequent request to the device's 1905.1 interface with the MAC address and receives either a rejection message back or no response from device's 1905.1 interface, then the ALME response in the resulting ALME-GET-INTF-PWR-STATE.response shall contain a reasonCode set to the value FAILURE.

5.1.6 ALME-GET-INTF-PWR-STATE.response

5.1.6.1 Function

This primitive is generated by the ALME as the response to an ALME-GET-INTF-PWR-STATE.request.

5.1.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
ALME-GET-INTF-PWR-STATE.response (
                                    intfAddress,
                                    powerState,
                                    reasonCode
                                )
```

Table 5-7 details the parameters for the ALME-GET-INTF-PWR-STATE.response.

Table 5-7—ALME-GET-INTF-PWR-STATE.response parameters

Name	Type	Valid range	Description
intfAddress	EUI-48 address	Any MAC address	The intfAddress provided in the ALME-GET-INTF-PWR-STATE.request
powerState	Enumeration	PWR_ON, PWR_SAVE, PWR_OFF	1905.1 interface power state
reasonCode	Enumeration	SUCCESS, FAILURE, UNMATCHED_MAC_ADDRESS	See Table 5-19

5.1.6.3 When generated

If the ALME receives an ALME-GET-INTF-PWR-STATE.request, then it shall respond with an ALME-GET-INTF-PWR-STATE.response.

The response shall indicate the SUCCESS of the ALME-GET-INTF-PWR-STATE.request with the powerState reported. It shall indicate the FAILURE of the ALME-GET-INTF-PWR-STATE.request based on interface nonresponse or rejection of request or UNMATCHED_MAC_ADDRESS when the MAC address in the ALME-GET-INTF-PWR-STATE.request is not listed in the ALME as an interface MAC address.

The powerState field is meaningful only for SUCCESS reasonCode.

5.1.6.4 Effect of receipt

The HLE is informed of the result of its ALME-GET-INTF-PWR-STATE.request.

5.1.7 ALME-SET-FWD-RULE.request

5.1.7.1 Function

This primitive is generated to add a new forwarding rule to the 1905.1 abstraction layer's optional forwarding entity. The forwarding database is the set of forwarding rules the 1905.1 AL forwarding entity used to determine the 1905.1 abstraction layer egress interface to which a given MSDU is forwarded.

5.1.7.2 Semantics of the service primitive

The primitive parameters are as follows:

ALME-SET-FWD-RULE.request (
classificationSet,
intfAddressList
)

Table 5-8 details the parameters for the ALME-SET-FWD-RULE.request.

Table 5-8—ALME-SET-FWD-RULE.request parameters

Name	Type	Valid range	Description
classificationSet		As defined in Table 5-9	Bit matching pattern of the rule
intfAddressList	EUI-48 Address(es)	Any MAC address	A list of the physical MAC addresses of the underlying network technology MACs to which the frames matching the classificationSet shall be forwarded

The intfAddressList consists of one or multiple physical MAC addresses of the underlying network technology MACs.

A classificationSet consists of elements detailed in Table 5-9.

Table 5-9—ClassificationSet elements

Name	Type	Valid range	Description
macDa	EUI-48 Address	Any MAC address	MAC destination address
macDaFlag	BOOLEAN	TRUE, FALSE	If FALSE, the MAC DA element is ignored
macSa	EUI-48 Address	Any MAC address	MAC source address
macSaFlag	BOOLEAN	TRUE, FALSE	If FALSE, the MAC SA element is ignored
etherType	Integer	As per IEEE Std 802-2001	etherType (see 10.4 of IEEE Std 802-2001)
etherTypeFlag	BOOLEAN	TRUE, FALSE	If FALSE, the etherType element is ignored
vid	Integer	As defined in 802.1Q frame format	IEEE 802.1Q VLAN ID
vidFlag	BOOLEAN	TRUE, FALSE	If FALSE, the VID element is ignored
pcp	Integer	As defined in IEEE 802.1Q frame format	IEEE 802.1Q priority code point field
pcpFlag	BOOLEAN	TRUE, FALSE	If FALSE, the PCP element is ignored

5.1.7.3 When generated

This primitive is generated to add a new forwarding rule for the 1905.1 abstraction layer.

5.1.7.4 Effect of receipt

This request sets the forwarding rule of the specified 1905.1 interface(s). The receiving ALME subsequently issues an ALME-SET-FWD-RULE.confirm primitive that reflects the results of the forwarding rule set request.

If the ALME receives an ALME-SET-FWD-RULE.request when it is unable to add a new rule to the 1905.1 abstraction layer forwarding rules, or if an identical classification set already exists in the existing forwarding rules, or if the MAC address does not match any 1905.1 interface MAC address within the 1905.1 device, then the ALME response in the resulting ALME-SET-FWD-RULE.confirm shall contain a reasonCode set to a value other than SUCCESS. If the new rule was added successfully, then the reasonCode shall be set to SUCCESS.

If the ALME receives an ALME-SET-FWD-RULE.request with a MAC address in the intfAddressList that does not match any 1905.1 interface MAC address, then the ALME response in the resulting ALME- SET-FWD-RULE.confirm shall contain a reasonCode set to the value UNMATCHED_MAC_ADDRESS.

If the ALME receives an ALME-SET-FWD-RULE.request with a classification set identical (i.e., all the parameters are identical) to a classification set that already exists in the forwarding rules, then the ALME response in the resulting ALME-SET-FWD-RULE.confirm shall contain a reasonCode set to the value DUPLICATE_CLASSIFICATION_SET.

5.1.8 ALME-SET-FWD-RULE.confirm

5.1.8.1 Function

This primitive confirms the addition of a new forwarding rule requested by an ALME-SET-FWD-RULE.request.

5.1.8.2 Semantics of the service primitive

The primitive parameters are as follows:

```
ALME-SET-FWD-RULE.confirm  (
                                    ruleId,
                                    reasonCode
                                )
```

Table 5-10 details the parameters for the ALME-SET-FWD-RULE.confirm.

Table 5-10—ALME-SET-FWD-RULE.confirm parameters

Name	Type	Valid range	Description
ruleId	Integer	Any ID value	Unique ID of the added forwarding rule
reasonCode	Enumeration	SUCCESS, UNMATCHED_MAC_ADDRESS, DUPLICATE_CLASSIFICATION_SET	See Table 5-19

5.1.8.3 When generated

If the ALME receives an ALME-SET-FWD-RULE.request, then it shall respond with an ALME-SET-FWD-RULE.confirm after the new rule has been added or the request is rejected. The description for the

reason codes are described in 5.1.7.4. The value in the ruleId field is valid only if the reasonCode is set to SUCCESS.

5.1.8.4 Effect of receipt

The HLE is informed of the result of its ALME-SET-FWD-RULE.request.

5.1.9 ALME-GET-FWD-RULES.request

5.1.9.1 Function

This primitive is used by HLEs to retrieve the forwarding rules stored in the forwarding database of a 1905.1 abstraction layer's forwarding entity.

5.1.9.2 Semantics of the service primitive

The primitive parameters are as follows:

ALME-GET-FWD-RULES.request (
)

5.1.9.3 When generated

This primitive is generated by the HLE as a result of actions outside the scope of this specification.

5.1.9.4 Effect of receipt

If the ALME receives the ALME-GET-FWD-RULES.resquest primitive, then the ALME shall generate an ALME-GET-FWD-RULES.response primitive.

5.1.10 ALME-GET-FWD-RULES.response

5.1.10.1 Function

This primitive is used to respond to the ALME-GET-FWD-RULES.request.

5.1.10.2 Semantics of the service primitive

The primitive parameters are as follows:

ALME-GET-FWD-RULES.response (
fwdRuleList
)

Table 5-11 details the parameters for the ALME-GET-FWD-RULES.response.

Table 5-11—ALME-GET-FWD-RULES.response parameters

Name	Type	Valid range	Description
fwdRuleList	A set of fwdRules	As defined in Table 5-12	The list of forwarding rules in the forwarding database of the 1905.1 abstraction layer's forwarding entity

Each fwdRuleList consists of the elements detailed in Table 5-12.

Table 5-12—fwdRuleList elements

Name	Type	Valid range	Description
ruleId	Integer	Any ID value	ID of the forwarding rule.
classificationSet		See Table 5-9	Classification set for the rule. See Table 5-9.
intfAddressList	EUI-48 Address(es)	Any MAC address	A list of the physical MAC addresses of the underlying network technology MACs to which frames matching the classificationSet shall be forwarded.
lastMatched	UInteger16	Any integer in the range [0,65 535]	The time interval (expressed in seconds) from the last time the classificationSet has been matched to the time the ALME-GET-FWD-RULES.request primitive has been issued. For instance, a value of 1 means that the classificationSet has been matched at least once within the last second. A value of 65 536 also covers time intervals greater than the maximum value measurable with the counter. A value of zero means that the information is not available.

The intfAddressList consists of one or multiple physical MAC addresses of the underlying network technology MACs.

The lastMatched information may be used by the HLE to decide the FWD rule to be removed in the presence of an ALME-SET-FWD-RULES.confirm primitive with reasonCode NBR_OF_FWD_ RULE_EXCEEDED.

This information may also be used by the HLE to remove a FWD rule when there is suspicion of incoherency between the forwarding rules of the optional 1905.1 abstraction layer's forwarding entity and the real topology of the network.

5.1.10.3 When generated

If the ALME receives an ALME-GET-FWD-RULES.request primitive, then the ALME shall generate an ALME-GET-FWD-RULES.response primitive.

5.1.10.4 Effect of receipt

The HLE is provided with the current forwarding rules of the 1905.1 abstraction layer's forwarding entity.

5.1.11 ALME-MODIFY-FWD-RULE.request

5.1.11.1 Function

This primitive is used by HLEs to modify a forwarding rule of the 1905.1 abstraction layer's forwarding entity.

5.1.11.2 Semantics of the service primitive

The primitive parameters are as follows:

ALME-MODIFY-FWD-RULE.request (

ruleId,
intfAddressList
)

Table 5-13 details the parameters for the ALME-MODIFY-FWD-RULE.request.

Table 5-13—ALME-MODIFY-FWD-RULE.request parameters

Name	Type	Valid range	Description
ruleId	Integer	The ID value of the fwdRuleList	Rule ID of the rule to modify.
intfAddressList	EUI-48 address(es)	Any MAC address	A list of the physical MAC addresses of the underlying network technology MACs to which frames the frames matching this forwarding rule shall be forwarded.

The intfAddressList consists of one or multiple physical EUI-48 addresses of the underlying network technology MACs.

5.1.11.3 When generated

This primitive is used by HLEs to modify a forwarding rule of the 1905.1 abstraction layer's forwarding entity.

5.1.11.4 Effect of receipt

This requests modification of the forwarding rule of the specified 1905.1 interface. The receiving ALME subsequently issues an ALME-MODIFY-FWD-RULE.confirm primitive that reflects the results of the forwarding rule modify request.

If the ALME receives an ALME-MODIFY-FWD-RULE.request in which the ruleId parameter has an invalid value, or if the MAC address does not match any 1905.1 interface MAC address, then the ALME response in the resulting ALME-MODIFY-FWD-RULE.confirm shall contain a reasonCode set to an appropriate value other than SUCCESS. Otherwise the intfAddressList of the ruleId is fully eliminated and replaced with the new intfAddressList provided by the ALME-MODIFY-FWD-RULE.request, and the ALME response in the resulting ALME-MODIFY-FWD-RULE.confirm shall contain a reasonCode set to the value SUCCESS.

5.1.12 ALME-MODIFY-FWD-RULE.confirm

5.1.12.1 Function

This primitive confirms the modification of the forwarding rule requested by an ALME-MODIFY-FWD-RULE.request primitive.

5.1.12.2 Semantics of the service primitive

The primitive parameters are as follows:

ALME-MODIFY-FWD-RULE.confirm (
ruleId,
reasonCode
)

Table 5-14 details the parameters for the ALME-MODIFY-FWD-RULE.confirm.

Table 5-14—ALME-MODIFY-FWD-RULE.confirm parameters

Name	Type	Valid range	Description
ruleId	Integer	Any ID value	Rule ID of the modified forwarding rule
reasonCode	Enumeration	SUCCESS, UNMATCHED_MAC_ADDRESS, INVALID_RULE_ID	See Table 5-19

5.1.12.3 When generated

If the ALME receives an ALME-MODIFY-FWD-RULE.request, then the ALME shall generate an ALME-MODIFY-FWD-RULE.confirm after the new rule has been modified or the modification is rejected.

5.1.12.4 Effect of receipt

The HLE is informed of the result of its ALME-MODIFY-FWD-RULE.request.

5.1.13 ALME-REMOVE-FWD-RULE.request

5.1.13.1 Function

This primitive is used to remove a forwarding rule to the 1905.1 abstraction layer.

5.1.13.2 Semantics of the service primitive

The primitive parameters are as follows:

ALME-REMOVE-FWD-RULE.request (
ruleId,
)

Table 5-15 details the parameters for the ALME-REMOVE-FWD-RULE.request.

Table 5-15—ALME-REMOVE-FWD-RULE.request parameters

Name	Type	Valid range	Description
ruleId	Integer	Any ID value of the fwdRuleList	Rule ID of the rule to remove

5.1.13.3 When generated

This primitive is used by HLEs to remove a forwarding rule of the 1905.1 abstraction layer's forwarding entity.

5.1.13.4 Effect of receipt

This requests removal of the forwarding rule of the specified 1905.1 interface. The receiving ALME subsequently issues an ALME-REMOVE-FWD-RULE.confirm primitive that reflects the results of the forwarding rule removal request.

If the ALME receives an ALME-SET-REMOVE-FWD-RULE.request in which the ruleId parameter has an invalid value, then the ALME response in the resulting ALME-REMOVE-FWD-RULE.confirm shall be set to the value INVALID_RULE_ID.

5.1.14 ALME-REMOVE-FWD-RULE.confirm

5.1.14.1 Function

This primitive confirms the deletion of the forwarding rule requested by an ALME-REMOVE-FWD-RULE.request.

5.1.14.2 Semantics of the service primitive

The primitive parameters are as follows:

```
ALME-REMOVE-FWD-RULE.confirm (
                                        ruleId,
                                        reasonCode
                                    )
```

Table 5-16 details the parameters for the ALME-REMOVE-FWD-RULE.confirm.

Table 5-16—ALME-REMOVE-FWD-RULE.confirm parameters

Name	Type	Valid range	Description
ruleId	Integer	Any ID value	Rule ID of the modified forwarding rule
reasonCode	Enumeration	SUCCESS, INVALID_RULE_ID	See Table 5-19

5.1.14.3 When generated

If the ALME receives an ALME-REMOVE-FWD-RULE.request, then the ALME shall generate an ALME-REMOVE-FWD-RULE.confirm after the new rule has been removed or the removal is rejected.

5.1.14.4 Effect of receipt

The HLE is informed of the result of its ALME-REMOVE-FWD-RULE.request.

5.1.15 ALME-GET-METRIC.request

5.1.15.1 Function

This primitive is used by an HLE to retrieve the link metric information of the transmission channel of one or all 1905.1 links from the list of the 1905.1 interfaces of the underlying network technology(ies).

5.1.15.2 Semantics of the service primitive

The primitive parameters are as follows:

```
ALME-GET-METRIC.request (
                                MACAddress,
                )
```

The MAC address is either the MAC address of a neighbor 1905.1 device or NULL.

5.1.15.3 When generated

This primitive is generated by the HLE to get the 1905.1 link metrics information on the links' transmission channel between the 1905.1 abstraction layer and the 1905.1 abstraction layer of its 1905.1 neighbor device(s).

5.1.15.4 Effect of receipt

If the ALME receives an ALME-GET-METRIC.request, then the ALME shall generate an ALME-GET-METRIC.response.

If the ALME-GET-METRIC.request contains a MAC address, then the ALME-GET-METRIC.response shall contain the 1905.1 link metrics for the 1905.1 links between the local 1905.1 interfaces and the 1905.1 interfaces of the specified 1905.1 neighbor device identified by the MAC address. If the ALME-GET-METRIC.request does not contain a MAC address, the ALME-GET-METRIC.response shall contain the 1905.1 link metrics for all the 1905.1 links between the 1905.1 interfaces of the local abstraction layer and the 1905.1 interfaces of all its 1905.1 neighbor devices.

If an HLE requests linkMetrics and metrics from the other end of a 1905.1 link (see Table 6-16) are needed to generate the linkMetrics for this given link, then the local 1905.1 management entity shall perform the 1905.1 link metric query/response procedure between the local 1905.1 device and the 1905.1 neighbor device using the procedures according to 7.4 and the message format described in 6.3.5.

If the ALME receives an ALME-GET-METRIC.request with a MAC address that does not match any of the device's neighbor 1905.1 abstraction layer MAC addresses, then the ALME response in the resulting ALME-GET-METRIC.response shall contain a reasonCode set to the value UNMATCHED_NEIGHBOR_MAC_ADDRESS.

5.1.16 ALME-GET-METRIC.response

5.1.16.1 Function

This primitive is used to send a response to the request to get the list of 1905.1 link metrics for the interfaces of the 1905.1 abstraction layer.

5.1.16.2 Semantics of the service primitive

The primitive parameters are as follows:

```
ALME-GET-METRIC.response (
                                    metricList,
                                    reasonCode
                          )
```

Table 5-17 details the parameters for the ALME-GET-METRIC.response.

Table 5-17—ALME-GET-METRIC.response parameters

Name	Type	Valid range	Description
metricList	A list of metricDescriptors	As defined in Table 5-18	The list of metricDescriptors
reasonCode	Enumeration	SUCCESS UNMATCHED_NEIGHBOR_MAC_ADDRESS	As defined in Table 5-19

Each metricDescriptor consists of the elements detailed in Table 5-18.

Table 5-18—metricDescriptor

Name	Type	Valid range	Description
neighborDevAddress	EUI-48 Address	Any MAC address	AL MAC address of the 1905.1 neighbor device associated with the 1905.1 link metrics
localIntfAddress	EUI-48 Address	Any MAC address	MAC address of the local interface associated with the 1905.1 link metrics
IEEE802.1BridgeFlag	Boolean	TRUE,FALSE	Boolean flag set as described in 8.1 to indicate that the 1905.1 neighbor device is connected on this particular interface: — Through one or more IEEE 802.1 bridges (TRUE) — Otherwise (FALSE)
linkMetrics		As defined in Table 6-17 and Table 6-19	The link metrics of the transmission channel of the 1905.1 link between the current 1905.1 device and a 1905.1 neighbor device

5.1.16.3 When generated

This primitive is generated by the 1905.1 ALME as a response to an ALME-GET-METRIC.request.

5.1.16.4 Effect of receipt

The receipt of this primitive provides an HLE with the 1905.1 link metrics of the 1905.1 interfaces connecting to a specified MAC address or the 1905.1 link metrics for all the MAC addresses of 1905.1 neighbor devices connected to the abstraction layer's 1905.1 interfaces.

Table 5-19—Encoding of reasonCode to reasonCode field values

reasonCode	reasonCode field value	Meaning
SUCCESS	0x00	
UNMATCHED_MAC_ADDRESS	0x01	The MAC address is not matched by any 1905.1 interface MAC address.
UNSUPPORTED_PWR_STATE	0x02	The requested power state transition is unsupported by the 1905.1 interface.
UNAVAILABLE_POWER_STATE	0x03	The requested power state transition is currently unavailable by the 1905.1 interface.
NBR_OF_FWD_RULE_EXCEEDED	0x04	No new rule could be added to the current set of forwarding rules.
INVALID_RULE_ID	0x05	Invalid rule ID value.
DUPLICATE_CLASSIFICATION_SET	0x06	An identical classificationSet already exists in the current set of forwarding rules.
UNMATCHED_NEIGHBOR_MAC_ADDRESS	0x07	The MAC address does not match any neighbor's 1905.1 AL MAC address.
FAILURE	0x10	The 1905.1 interface has either rejected the request or is nonresponsive.

5.2 AL data (MSDU) services (informative)

For MSDUs, the IEEE 1905.1 abstraction layer supports the following service primitives as defined in ISO/IEC 8802-2-1998 [B2]:

— MA-UNITDATA.request
— MA-UNITDATA.indication

The LLC definitions of the primitives and specific parameter value restrictions imposed by the underlying network technologies are given by the network specific specifications listed below:

a) Subclause 2.3 of IEEE Std 802.3-2008
b) Subclause 5.2 of IEEE Std 802.11 -2012
c) Subclause 5.2 of IEEE Std 1901-2010
d) Subclause 5.1 of MoCA MAC/PHY Specification v1.1

5.2.1 MA-UNITDATA.request

5.2.1.1 Function

This primitive requests a transfer of an MSDU from the LLC entity to the 1905.1 abstraction layer entity.

5.2.1.2 Semantics of the service primitive

MA-UNITDATA.request ()

5.2.1.3 When generated

This primitive is generated by the LLC entity when an MSDU is to be transferred to the 1905.1 abstraction layer entity.

5.2.1.4 Effect of receipt

On receipt of this primitive, the abstraction layer entity passes the primitive's parameters to the 1905.1 interface(s) according to the forwarding rules of the 1905.1 abstraction layer's forwarding entity.

5.2.2 MA-UNITDATA.indication

5.2.2.1 Function

This primitive defines the transfer of an MSDU from the 1905.1 abstraction layer entity to the LLC entity.

5.2.2.2 Semantics of the service primitive

The parameters of the primitive are as follows:

MA-UNITDATA.indication()

The meaning of the parameters is a function of the underlying network technology of the 1905.1 interface from which the MSDU was received. If a parameter is not applicable to the underlying network technology, then its value is ignored.

5.2.2.3 When generated

The MA-UNITDATA.indication primitive is passed from the 1905.1 abstraction layer entity to the LLC to indicate the arrival of an MSDU received without error and forwarded to the LLC entity.

5.2.2.4 Effect of receipt

The effect of receipt of this primitive by the LLC is dependent on the content of the MSDU.

5.3 Informaion elements

5.3.1 VendorSpecificInfo information elements

The format of the VendorSpecificInfo information element (IE) is described in Table 5-20.

Table 5-20—VendorSpecificInfo information element

Field	Starting octet number	Field size (octets)	Type	Description
ieType	0	2	Enumeration	The value of the IE type field is 1 <<VendorSpecificInfo>>.
lengthField	2	2	Integer	The value of the length is *n* + 3.
oui	4	3	Integer	The value of the 24-bit globally unique IEEE-RA assigned number to the vendor.
vendorSpecificInfo	7	*n*		The vendor specific content contains vendor specific field(s). The length of the vendor specific content is limited by the maximum allowed MPDU size.

6. Interabstraction layer message formats

6.1 IEEE 802.1 bridge discovery message (neighbor multicast) format

An IEEE 802.1 bridge discovery message is used to assess if one or more IEEE 802.1 bridges exist between the transmitter and the receiver of the message. A link layer discovery protocol (LLDP) message is used for this purpose.

Table 6-1 is used for the IEEE 802.1 bridge discovery message.

Table 6-1—Ethernet frame header information

Field	Length	Value range	Description
macDa	6 octets	01-80-C2-00-00-0E	Nearest bridge group MAC address.
macSa	6 octets	Any EUI-48 value	Set to either the MAC address of the interface or the 1905.1 AL MAC address of the 1905.1 device from which a 1905.1 topology discovery message is transmitted. The choice of which MAC address to use is implementation specific.
etherType	2 octets	0x88CC	88-CC. LLDP EtherType.
payload	46 – 1500 octets	Any integer value	Link layer discovery protocol data unit (LLDPDU) (see IEEE Std 802.1AB™-2009).

The following LLDP type length values (TLVs) are used to form the LLDP message:

— Chassis ID TLV:
    1) Chassis ID subtype is set to 4—MAC address
    2) Chassis ID is set to the 1905.1 AL MAC address
— Port ID TLV:
    1) Port ID subtype is set to 3 (MAC address [IEEE Std 802-2001])
    2) Port ID is set to the MAC address of the interface on which this message is transmitted
— Time to live TLV:
    1) TTL is set to 180 s
— End of LLDPDU TLV

6.2 1905.1 CMDU

The 1905.1 CMDU is used to carry 1905.1 protocol TLVs from a transmitting 1905.1 device to one or more receiving 1905.1 devices (depending on whether the destination address is a unicast address or a group address). If the message is too large to fit within an Ethernet frame, then multiple fragments can be created at the TLV boundaries to form multiple messages (see 7.1).

When the 1905.1 management entity generates a CMDU, the following criteria must be met:

- It shall include all of the TLVs that are listed for the message.
- It shall not include any other TLV that is not listed for the message.
- It may additionally include zero or more vendor specific TLVs.

When the 1905.1 management entity receives a CMDU, the following criteria must be met:

- It may process or ignore any vendor specific TLVs.
- It shall ignore all TLVs that are not specified for the message.
- It shall ignore the entire message if the message does not include all of the TLVs that are listed for this message.

If multiple TLVs of the same type are included in one or more fragments of a message, then the 1905.1 management entity shall consider them as one TLV containing the aggregate of the tlvValue contents.

The byte ordering of all the messages are in big endian and the bit ordering is shown in Figure 6-1.

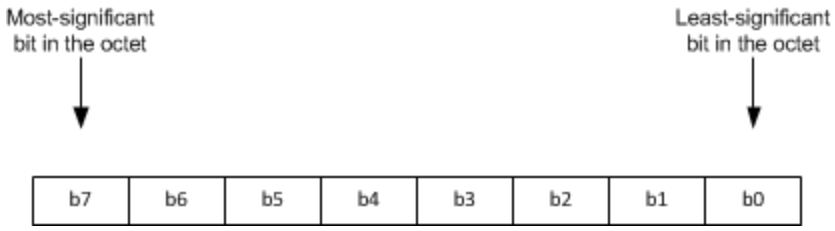

Figure 6-1—Bit ordering of an octet

6.2.1 Ethernet frame header

Table 6-2 and Table 6-3 shall be used for the Ethernet frame on which the message is carried.

Table 6-2—Ethernet frame header information

Field	Length	Value range	Description
macDa	6 octets	Any EUI-48 value	For unicast messages, this field shall be set to the 1905.1 AL MAC address of the receiving 1905.1 device; if the 1905.1 management entity of the sending device knows that the receiving 1905.1 interface is an IEEE 802.11 STA, then it may instead set this field to the MAC address of the receiving 1905.1 interface. For multicast messages, this field shall be set to the 1905.1 multicast MAC address for multicast messages.
macSa	6 octets	Any EUI-48 value	Either the AL MAC address or the MAC address of the interface from which the message is transmitted.
etherType	2 octets	0x893A	1905.1 EtherType.

Table 6-3—1905.1 CMDU

Field	Length	Value	Description
messageVersion	1 octet	0x00	Message version. 0x00: for this version of the specification 0x01~0xFF: reserved values.
reservedField	1 octet		All values are reserved.
messageType	2 octets		See Table 6-4.
message Id	2 octets		Identifies the message (see 7.8).
fragmentId	1 octet		Identifies the fragment of a message (see 7.1.1).
lastFragmentIndicator	1 bit (bit 7)		"1": last fragment "0": not last fragment
relayIndicator	1 bit (bit 6)		Indicate if the message shall: "1": be relayed (relayed multicast) subject to the rules in 7.6 "0": not be relayed (neighbor multicast or unicast)
reservedField	6 bits (bits 5 to 0)		All values are reserved.
1905.1Protocoltlvs	Variable length		TLV(s) (see 6.4).
endOfMessageTlv	3 octets		End of message TLV (see Table 6-6).

Table 6-4—Message type

Message type	Protocol	Value	Transmission type	Relay indicator field	Description
Topology discovery message	Topology discovery (see Clause 8)	0x0000	Neighbor multicast	0	A message to advertise a device's existence
Topology notification message	Topology discovery	0x0001	Relayed multicast	1	A message to notify that a device's 1905.1 topology entries have changed
Topology query message	Topology discovery	0x0002	Unicast	0	A message to query a device's topology information
Topology response message	Topology discovery	0x0003	Unicast	0	A message to carry topology information in response to a topology query
Vendor specific message	N/A	0x0004	Unicast/ Neighbor multicast/ relayed multicast	Vendor specific [0|1]	A message that is vendor specific
Link metric query message	Link metric information dissemination protocol (see 11.1)	0x0005	Unicast	0	A message to query the link metric information of a 1905.1 link between a specific device pair
Link metric response message	Link metric information dissemination protocol	0x0006	Unicast	0	A message to carry the link metric information in response to a link metric query
AP-autoconfiguration search message	AP-autoconfiguration protocol (see Clause 10)	0x0007	Relayed multicast	1	A message to search for a registrar
AP-autoconfiguration response message	AP-autoconfiguration protocol	0x0008	Unicast	0	A message to answer to a search message
AP-autoconfiguration Wi-Fi simple configuration (WSC) message	AP-autoconfiguration protocol	0x0009	Unicast	0	A message to carry a WSC registration frame
AP-autoconfiguration renew message	AP-autoconfiguration protocol	0x000A	Relayed multicast	1	A message to advertise a renewing registration is required for a specific band
1905.1 push button event notification message	Push button	0x000B	Relayed multicast	1	Advertise a push button event
1905.1 push button join notification message	Push button	0x000C	Relayed multicast	1	Advertise a successful join due to a push button event
N/A	N/A	0x000D~ 0xFFFF	N/A	N/A	Reserved

6.3 1905.1 message formats

This subclause defines the message formats for each message.

6.3.1 Topology discovery message format

The following TLVs shall be included in this message:

- One 1905.1 AL MAC address type TLV (see Table 6-8)
- One 1905.1 MAC address type TLV (see Table 6-9)

6.3.2 Topology query message format

No TLVs are required in this message.

6.3.3 Topology response message format

The following TLVs shall be included in this message:

- One device information type TLV (see Table 6-10)
- Zero or more device bridging capability TLVs (see Table 6-11)
  - If a 1905.1 device has more than one interface, then the 1905.1 management entity shall include one device bridging capability TLV.
- Zero or more non-1905 neighbor device list TLVs (see Table 6-14)
  - If a 1905.1 management entity infers the presence of a non-1905.1 neighbor device (see 8.1), then it shall include that device in the non-1905.1 neighbor device list TLV.
- Zero or more 1905.1 neighbor device TLVs (see Table 6-15)
  - If a 1905.1 management entity infers the presence of a 1905.1 neighbor device (see 8.1), then it shall include that device in the 1905.1 neighbor device TLV.

6.3.4 Topology notification message format

The following TLVs shall be included in this message:

- One 1905.1 AL MAC address type TLV (see Table 6-8)

6.3.5 Link metric query message format

The following TLVs shall be included in this message:

- One link metric query TLV (see Table 6-17)

6.3.6 Link metric response message format

Table 6-5 shows the TLVs that shall be included for each valid 1905.1 neighbor device, as requested in the link metric query message.

Table 6-5—Link metric response TLVs

Link metric requested	TLVs to be included
Tx link metrics only	Transmitter link metric TLV (see Table 6-17)
Rx link metrics only	Receiver link metric TLV (see Table 6-19)
Both Tx and Rx link metrics	Transmitter link metric TLV (see Table 6-17) and receiver link metric TLV (see Table 6-19)

If the specified neighbor 1905.1 AL ID does not identify a neighbor of the receiving 1905.1 AL, then a link metric ResultCode TLV (see Table 6-21) with a value set to "invalid neighbor" shall be included in this message.

6.3.7 AP-autoconfiguration search message format

The AP-autoconfiguration search message is a relayed multicast message.

The following TLVs shall be included in this message:

— One 1905.1 AL MAC address type TLV (see Table 6-8)
— One SearchedRole TLV (see Table 6-22)
— One AutoconfigFreqBand TLV (see Table 6-23)

6.3.8 AP-autoconfiguration response message format

The AP-autoconfiguration response message is a unicast message sent back in response to an AP-autoconfiguration search message.

The following TLVs shall be included in this message:

— One SupportedRole TLV (see Table 6-24)
— One SupportedFreqBand TLV (see in Table 6-25)

6.3.9 AP-autoconfiguration WSC message format

The AP-autoconfiguration WSC message is a unicast message used between 1905.1 devices to carry a WSC frame.

The following TLVs shall be included in this message:

— One WSC TLV (see Table 6-26)

6.3.10 AP-autoconfiguration renew message format

The AP-autoconfiguration renew message is a relayed multicast message.

The following TLVs shall be included in this message:

— One 1905.1 AL MAC address type TLV (see Table 6-8)

— One SupportedRole TLV (see Table 6-24)
— One SupportedFreqBand TLV (see in Table 6-25)

6.3.11 1905.1 push button event notification message format

The 1905.1 push button event notification is a 1905.1 relayed multicast message.

The following TLV shall be included in this message:

— One 1905.1 AL MAC address type TLV (see Table 6-8)
— One Push_Button_Event notification TLV (see Table 6-27)

6.3.12 1905.1 push button join notification message format

The 1905.1 push button join notification is a 1905.1 Relayed Multicast message.

The following TLVs shall be included in this message:

— One 1905.1 AL MAC address type TLV (see Table 6-8)
— One Push_Button_Join notification TLV (see Table 6-28)

6.3.13 Vendor specific message format

If the relay indicator in this message is set to "0," then this message is a unicast message or a neighbor multicast message. Otherwise, this message is a relayed multicast message.

The following TLV shall be included in this message

— The vendor specific TLV (see Table 6-7) as the first TLV followed by zero or more TLVs (either TLVs defined in this specification or any vendor specific TLVs).

6.4 1905.1 TLVs

This subclause defines the TLVs for each message.

tlvType is a 1-octet field that indicates the type of TLV.

tlvLength is a 2-octet field where the 2 most significant bits are reserved and the 14 least significant bits indicate the length in number of octets of the tlvValue field (excluding the tlvType and tlvLength fields).

tlvValue is a variable-length field that indicates the value carried by the TLV.

6.4.1 End of message TLV

Table 6-6 describes the end of message TLV.

Table 6-6—End of message TLV

Field	Length	Value	Description
tlvType	1 octet	0x00	End of message TLV (TLV indicating the end of message)
tlvLength	2 octets	0x0000	Null

6.4.2 Vendor specific TLV

Table 6-7 describes the vendor specific TLV.

Table 6-7—Vendor specific TLV

Field	Length	Value	Description
tlvType	1 octet	11	Vendor specific TLV
tlvLength	2 octets	3 + m	Number of octets in ensuing field
tlvValue	3 octets		Vendor specific OUI, the value of the 24-bit globally unique IEEE-SA assigned number to the vendor
	m octets		Vendor specific information

6.4.3 1905.1 AL MAC address type TLV

Table 6-8 describes the 1905.1 AL MAC address type TLV.

Table 6-8—1905.1 AL MAC address type TLV

Field	Length	Value	Description
tlvType	1 octet	1	1905.1 AL MAC address type TLV
tlvLength	2 octets	6	Number of octets in ensuing field
tlvValue	6 octets	Any EUI-48 value	1905.1 AL MAC address of the transmitting device

6.4.4 MAC address type TLV

Table 6-9 describes the MAC address type TLV.

Table 6-9—MAC address type TLV

Field	Length	Value	Description
tlvType	1 octet	2	MAC address type TLV
tlvLength	2 octets	6	Number of octets in ensuing tlvValue field
tlvValue	6 octets	Any EUI-48 value	MAC address of the interface on which the message is transmitted

6.4.5 1905.1 device information type TLV

Table 6-10 describes the 1905.1 device information type TLV.

Table 6-10—1905.1 device information type TLV

Field	Length	Value	Description
tlvType	1 octet	3	Device information type.
tlvLength	2 octets	Variable	Number octets in ensuing field.
tlvValue	6 octets		1905.1 AL MAC address of the device.
	1 octet	*k*	Number of local interfaces.
	6 octets	Any EUI-48 value	MAC address of the local interface.
	2 octets		Media type of the local interface (as defined in the "Media type" and "Description" columns of Table 6-12).
	1 octet	n	Number of octets in ensuing field.
	n octets		Media-specific information of the local interface (as defined in the "Media-specific information" column of Table 6-12.
			The above four fields are repeated $k - 1$ times.

6.4.6 Device bridging capability TLV

Table 6-11 describes the device bridging capability TLV (meaningful when MSDU could be forwarded between 1905.1 interfaces).

Table 6-11—Device bridging capability TLV

Field	Length	Value	Description
tlvType	1 octet	4	Device bridging capability.
tlvLength	2 octets	Variable	Number of octets in ensuing field.
tlvValue	1 octet	*m*	*m* is the total number of bridging tuples in this TLV.
	1 octet	*k*	Number of MAC addresses in this bridging tuple.
	6 octets	Any EUI-48 value	The MAC address of a 1905.1 device's network interface that belongs to this bridging tuple.
			The above field is repeated $k - 1$ times.
			The above two fields are repeated $m - 1$ times.

6.4.7 Media type

Table 6-12 describes the media type.

Table 6-12—Media type (intfType)

Media type (intfType)		Description	Media-specific information (*n* octets)
(Bits 15 to 8)	(Bits 7 to 0)		
0	0	IEEE 802.3u fast Ethernet	N/A (*n* = 0)
	1	IEEE 802.3ab gigabit Ethernet	N/A (*n* = 0)
	2 to 255	Reserved values	Reserved (*n* = 0)
1	0	IEEE 802.11b (2.4 GHz)	*n* = 10 (see Table 6-13)
	1	IEEE 802.11g (2.4 GHz)	
	2	IEEE 802.11a (5 GHz)	
	3	IEEE 802.11n (2.4 GHz)	
	4	IEEE 802.11n (5 GHz)	
	5	IEEE 802.11ac (5 GHz)	
	6	IEEE 802.11ad (60 GHz)	
	7	IEEE 802.11af (whitespace)	
	8 to 255	Reserved values	Reserved (*n* = 0)
2	0	IEEE 1901 wavelet	Network membership: network identifier *n* =7
	1	IEEE 1901 FFT	
	2 to 255	Reserved values	Reserved
3	0	MoCA v1.1	*n* = 0
	1 to 255	Reserved values	
4 to 254	0 to 255	Reserved values	Reserved (*n* = 0)
255	255	Unknown media	*n* = 0

Table 6-13 describes the IEEE 802.11 specific information.

Table 6-13—IEEE 802.11 specific information

Field	Length	Value range	Description
networkMembership	6 octets	Any integer value	BSSID
role	4 bits (Bits 7 to 4)	Any integer value	"0000" – AP "0001" – "0011" Reserved "0100" – non-AP/non-PCP STA "1000" – Wi-Fi P2P Client (see [B04]) "1001" – Wi-Fi P2P Group Owner (see [B04]) "1010" – 802.11adPCP "1011" – "1111" Reserved
Reserved	4 bits (Bits 3 to 0)	All zeroes	Reserved
apChannelBand	1 octet	Any integer value	Hex value of dot11CurrentChannelBandwidth (see IEEE P802.11ac/D3.0 for description)
apChannelCenterFrequencyIndex1	1 octet	Any integer value	Hex value of dot11CurrentChannelCenterFrequencyIndex1 (see IEEE P802.11ac/D3.0 for description)
apChannelCenterFrequency Index2	1 octet	Any integer value	Hex value of dot11CurrentChannelCenterFrequencyIndex2 (see IEEE P802.11ac/D3.0 for description)

6.4.8 Non-1905 neighbor device list TLV

Table 6-14 describes the non-1905 neighbor device list TLV.

Table 6-14—Non-1905 neighbor device list TLV

Field	Length	Value	Description
tlvType	1 octet	6	List of connected non-1905 neighbor devices.
tlvLength	2 octets	Variable	Number of octets in ensuing field.
tlvValue	6 octets	Any EUI-48 value	MAC address of the local interface.
	6 octets	Any EUI-48 value	MAC address of non-1905 neighbor device.
			The above field is repeated 0 or more times.

6.4.9 1905.1 neighbor device TLV

Table 6-15 describes the 1905.1 neighbor device TLV.

Table 6-15—1905.1 neighbor device TLV

Field	Length	Value range	Description
tlvType	1 octet	7	Information on 1905.1 neighbor device.
tlvLength	2 octets	Any integer value	Number of octets in ensuing field.
tlvValue	6 octets	Any EUI-48 value	MAC address of the local interface.
	6 octets	Any EUI-48 value	1905.1 AL MAC address of 1905.1 neighbor.
	1 bit (Bit 7)	0 or 1	Existence of IEEE 802.1 bridges: 0: no IEEE 802.1 bridges exist. 1: at least one IEEE 802.1 bridge exists between this device and the neighbor.
	7 bits (Bits 6 to 0)	All zeroes	Reserved.
			The above three fields are repeated 0 or more times.

6.4.10 Link metric query TLV

Table 6-16 describes the link metric query TLV.

Table 6-16—Link metric query TLV

Field	Length	Value range	Description
tlvType	1 octet	8	Link metric.
tlvLength	2 octets	8	Number of octets in ensuing field.
tlvValue	1 octet	0x00: All neighbors 0x01: Specific neighbor 0x02 ~ 0xFF: Reserved values	If the value is 0, then the EUI-48 field is not present; if the value is 1, then the EUI-48 field shall be present.
	6 octets	any EUI-48 value	1905.1 AL MAC address of a neighbor of the receiving device.
	1 octet	0x00 : Tx link metrics only 0x01 : Rx link metrics only 0x02 : Both Tx and Rx link metrics 0x03~0xFF: Reserved values	The link metrics requested.

6.4.11 1905.1 transmitter link metric TLV

Table 6-17 describes the 1905.1 transmitter link metric TLV.

Table 6-17—1905.1 transmitter link metric TLV

Field	Length	Value range	Description
tlvType	1 octet	9	Transmitter link metric.
tlvLength	2 octets	$12 + 29 \times n$	Number of octets in ensuing field. *n* can be one or more.
tlvValue	6 octets	Any EUI-48 value	1905.1 AL MAC address of the device that transmits the response message that contains this TLV.
	6 octets	Any EUI-48 value	1905.1 AL MAC address of the neighbor whose link metric is reported in this TLV.
	The following fields shall be repeated for each connected interface pair between the receiving 1905.1 AL and the neighbor 1905.1 AL.		
tlvValue	6 octets	Any EUI-48 value	MAC address of an interface in the receiving 1905.1 AL, which connects to an interface in the neighbor 1905.1 AL.
tlvValue	6 octets	Any EUI-48 value	MAC address of a 1905.1 interface in a neighbor 1905.1 device, which connects to a 1905.1 interface in the receiving 1905.1 device.
tlvValue	17 octets	See Table 6-18	Link metric information for the above interface pair between the receiving 1905.1 AL and the neighbor 1905.1 AL. Format follows Table 6-18.

Table 6-18 describes the 1905.1 transmitter link metrics.

Table 6-18—1905.1 transmitter link metrics

Name	Type	Length	Value range	Description
intfType	Enumeration	2 octets	As defined in Table 6-12.	The underlying network technology.
IEEE802.1BridgeFlag	Boolean	1 octet	0x00: Indicates that the 1905.1 link does not include an IEEE 802.1 bridge. 0x01: Indicates that the 1905.1 link includes one or more IEEE 802.1 bridges. 0x02~0xFF Reserved values.	Indicates whether or not the 1905.1 link includes one or more IEEE 802.1 bridges.
packetErrors	Integer32	4 octets	Any integer value	Estimated number of lost packets on the transmit side of the link during the measurement period.
transmittedPackets	Integer32	4 octets	Any integer value	Estimated number of packets transmitted by the Transmitter of the link on the same measurement period used to estimate packetErrors.
macThroughputCapacity	Integer16	2 octets	Any integer value	The maximum MAC throughput of the Link estimated at the transmitter and expressed in Mb/s.[18]
linkAvailability	Integer16	2 octets	Integer value between 0 and 100	The estimated average percentage of time that the link is available for data transmissions.
phyRate	Integer 16	2 octets	Any integer value	If the media type of the link is IEEE 802.3, then IEEE 1901 or MoCA 1.1 (8 MSB bits value of media type as defined in Table 6-12 is 0, 2, or 3). This value is the PHY rate estimated at the transmitter of the link expressed in Mb/s; otherwise, it is set to 0xFFFF.

All measurements in Table 6-18 are made at the 1905.1 interface level.

6.4.12 1905.1 receiver link metric TLV

Table 6-19 describes the 1905.1 receiver link metric TLV.

[18] The MAC throughput capacity is a function of the PHY rate and of the MAC overhead.

Table 6-19—1905.1 receiver link metric TLV

Field	Length	Value range	Description
tlvType	1 octet	10	Receiver link metric.
tlvLength	2 octets	$12 + 23 \times N$	Number of octets in ensuing field *N* can be one or more.
tlvValue	6 octets	Any EUI-48 value	1905.1 AL MAC address of the device that transmits the response message that contains this TLV.
	6 octets	Any EUI-48 value	1905.1 AL MAC address of a neighbor of the neighbor whose link metric is reported in this TLV.
	The following fields shall be repeated for each connected interface pair between the receiving 1905.1 AL and the neighbor 1905.1 AL.		
	6 octets	Any EUI-48 value	MAC address of an interface in the receiving 1905.1 AL that connects to an interface in the neighbor 1905.1 AL.
	6 octets	Any EUI-48 value	MAC address of an interface in neighbor 1905.1 AL that connects to an interface in the receiving 1905.1 AL.
	11 octets		Link metric information for the above interface pair between the receiving 1905.1 AL and the neighbor 1905.1 AL. The format follows Table 6-20.

Table 6-20 describes the 1905.1 receiver link metrics.

Table 6-20—1905.1 receiver link metrics

Name	Type	Length	Value range	Description
intfType	Enumeration	2 octets	As defined in Table 6-12	Identifies the underlying media type.
packetErrors	Integer32	4 octets	Any integer value	Estimated number of lost packets during the measurement period.
packetsReceived	Integer32	4 octets	Any integer value	Number of packets received at the interface during the same measurement period used to count packetErrors.
rssi	Integer 8	1 octet	Any integer value	If the media type of the link is IEEE 802.11 (8 MSB value of media type as defined in Table 6-12, then this value is the estimated RSSI in dB at the receive side of the Link expressed in dB; otherwise, it is set to 0xFF.

6.4.13 1905.1 link metric result code TLV

Table 6-21 describes the 1905.1 link metric result code TLV.

Table 6-21—1905.1 link metric result code TLV

Field	Length	Value range	Description
tlvType	1 octet	12	Result code TLV
tlvLength	2 octets	1	Number of octets in ensuing field
tlvValue	1 octet	0x00: Invalid neighbor 0x01 ~ 0xFF: Reserved values	Result code for the link metric query message

6.4.14 SearchedRole TLV

Table 6-22 describes the SearchedRole TLV.

Table 6-22—SearchedRole TLV

Field	Length	Value range	Description
tlvType	1 octet	13	SearchedRole
tlvLength	2 octets	1	Number of octets in ensuing field
tlvValue	1 octet	0x00: Registrar 0x01~0xFF: Reserved values	Type of role of the unconfigured interface requesting an autoconfiguration

6.4.15 AutoconfigFreqBand TLV

Table 6-23 describes the AutoconfigFreqBand TLV.

Table 6-23—AutoconfigFreqBand TLV

Field	Length	Value range	Description
tlvType	1 octet	14	Unconfigured frequency band
tlvLength	2 octets	1	Number of octets in ensuing field
tlvValue	1 octet	0x00: 802.11 2.4 GHz 0x01: 802.11 5 GHz 0x02: 802.11 60 GHz 0x03~0xFF: Reserved values	Frequency band of the unconfigured interface requesting an autoconfiguration

6.4.16 SupportedRole TLV

Table 6-24 describes the SupportedRole TLV.

Table 6-24—SupportedRole TLV

Field	Length	Value range	Description
tlvType	1 octet	15	SupportedRole
tlvLength	2 octets	1	Number of octets in ensuing field
tlvValue	1 octet	0x00: Registrar 0x01~0xFF: Reserved values	Type of role

6.4.17 SupportedFreqBand TLV

Table 6-25 describes the SupportedFreqBand TLV.

Table 6-25—SupportedFreqBand TLV

Field	Length	Value range	Description
tlvType	1 octet	16	Supported frequency band
tlvLength	2 octets	1	Number of octets in ensuing field
tlvValue	1 octet	0x00: 802.11 2.4 GHz 0x01: 802.11 5 GHz 0x02: 802.11 60 GHz 0x03~0xFF: Reserved values	Frequency band supported by the autoconfiguration process

6.4.18 WSC TLV

Table 6-26 describes the WSC TLV.

Table 6-26—WSC TLV

Field	Length	Value range	Description
tlvType	1 octet	17	WSC Type
tlvLength	2 octets	*n*	Number of octets in ensuing field
tlvValue	*n* octets	WSC frame	The WSC frame (either M1 or M2)

6.4.19 Push_Button_Event notification TLV

Table 6-27 describes the Push_Button_Event notification TLV.

Table 6-27—Push_Button_Event notification TLV

Field	Length	Value range	Description
tlvType	1 octet	18	Push button event notification TLV.
tlvLength	2 octets	Variable	Number of octets in ensuing field.
tlvValue	1 octet	Any integer value *n*	Number of media types included in this message: can be 0 or larger.
	2 octets	Media type (intfType) as defined in Table 6-12	A media type for which a push button configuration method has been activated on the device that originates the push button event notification. If *n* is 0, then this field is not included.
	1 octet	*k*	Number of octets in ensuing field.
	k octets	Media-specific information as defined in Table 6-12	Media-specific information corresponding to the media indicated in the previous field. If *k* is 0, then this field is not included.
			The previous three fields are repeated *n* – 1 times (or none if *n* is 0).

6.4.20 Push_Button_Join notification TLV

Table 6-28 describes the Push_Button_Join notification TLV.

Table 6-28—Push_Button_Join notification TLV

Field	Length	Value range	Description
tlvType	1 octet	19	Push button join notification TLV
tlvLength	2 octet	20	Number of octets in ensuing field
tlvValue	6 octets	Any EUI-48 value	The AL ID of the device that sent the push button event notification message
	2 octets	Any integer value	The message identifier (MID) of the push button event notification message
	6 octets	Any EUI-48 value	Interface-specific MAC address of the interface of the transmitting device belonging to the medium on which a new device joined.
	6 octets	Any EUI-48 value	Interface-specific MAC address of the interface of the new device that was joined to the network as a result of the push button configuration sequence.

7. IEEE 1905.1 protocol rules/procedures

7.1 Fragmentation and reassembly of a control message data unit (CMDU)

If a transmitting 1905.1 device determines it would form a CMDU that exceeds 1500 octets, then it shall fragment the CMDU into multiple fragments for transmission according to this subclause.

7.1.1 Fragmentation procedures

If the transmitting device forms a CMDU whose size exceeds 1500 octets, then the transmitter shall fragment the CMDU into multiple CMDU fragments for transmission. Each CMDU fragment may carry a partial payload of the original CMDU, fragmented at the 1905.1 protocol TLV boundaries. The transmitting device shall generate CMDU fragments where each fragment contains one or more 1905.1 protocol TLVs such that the length of each CMDU fragment does not exceed 1500 octets. During fragmentation, each successively generated CMDU fragment carries the same MID, as well as a fragment identifier (FID) with assigned values monotonically increasing by 1, starting from 0 for the first CMDU fragment. The transmitting device shall set the last fragment indicator field to “1” in the last CMDU fragment and to “0” in all other fragments. An unfragmented CMDU shall set the FID to “0” and the last fragment indicator field to “1.”

7.1.2 Reassembly procedures

If the receiving device receives a CMDU fragment, then it shall attempt to reassemble all of the fragments of the original CMDU. The receiving device shall forward the reassembled CMDU to the upper layer only if all the fragments have been received (the last fragment indicator identifies the last fragment). The timeout for expected (e.g., fragmented) CMDUs is implementation dependent.

7.2 CMDU neighbor multicast transmission procedures

A 1905.1 management entity shall transmit a neighbor multicast CMDU by:

— Transmitting the CMDU once on each and every of its authenticated 1905.1 interfaces that would be reported in the “device information type TLV” that are in the state of “PWR_ON” or “PWR_SAVE”

7.3 CMDU relayed multicast transmission procedures

A 1905.1 management entity shall transmit a relayed multicast CMDU by:

— Transmitting the CMDU exactly once on each and every of its authenticated 1905.1 interfaces that would be reported in the "device information type TLV" that are in the state of "PWR_ON" or "PWR_SAVE"

7.4 CMDU unicast transmission procedures

In the CMDU unicast transmission procedure, a 1905.1 management entity shall transmit a unicast CMDU by:

— Selecting at least one of the authenticated 1905.1 interfaces connected to a 1905.1 device addressed by the CMDU's destination address, where the powerState of the device is at "PWR_ON" or "PWR_SAVE"
— Transmitting the CMDU exactly once on the selected interface(s)

7.5 CMDU neighbor multicast reception procedures

If a 1905.1 management entity receives a neighbor multicast CMDU that it has not previously received, and that it has not generated, then it shall:

— Process the CMDU
— Not transmit the CMDU on any of its interfaces

A 1905.1 management entity can use the 1905.1 AL MAC address type TLV and MID tuple to determine whether it has previously generated or received a CMDU.

7.6 CMDU relayed multicast reception procedures

If a 1905.1 management entity receives a relayed multicast CMDU that it has not previously received and that it has not generated, and in which the "relay indicator" is set in the CMDU, then it shall:

— Process the CMDU
— Retransmit the CMDU exactly once on each of its interfaces on which it did not receive the CMDU

A 1905.1 management entity shall not retransmit any CMDU that it has previously transmitted. A 1905.1 management entity can use the 1905.1 AL MAC address type TLV and MID tuple to determine whether it has previously generated or received a message.

7.7 CMDU unicast reception procedures

If a 1905.1 management entity receives a unicast CMDU that it has not previously received and that it has not generated, then it shall:

— Process the CMDU
— Not transmit the CMDU on any of its interfaces

A 1905.1 management entity can use the 1905.1 AL MAC address type TLV and MID tuple to determine whether it has previously generated or received the CMDU.

7.8 Message identifier values

1905.1 messages include a MID field. The MID of a CMDU, depending on its message type, shall be set to the same value as that of a received CMDU or to a new value.

If a 1905.1 management entity generates a message whose requirements mandate a new MID value, then it shall set the new MID value to the previously generated new MID value incremented by one, modulo 2^{16}.

7.9 Reserved values, fields, and bits

A 1905.1 management entity shall not transmit reserved values.

A 1905.1 management entity shall set reserved bit fields to zero on transmission and shall ignore them on receipt.

If a 1905.1 management entity receives a message with "reserved values" in any of the fields in the 1905.1 CMDU, then it shall:

- If the relay indicator is set to "1," relay the message according to 7.6 and ignore the rest of the message.
- Discard and ignore the entire 1905.1 message.

If a 1905.1 management entity receives a message with "reserved values" in any field in a TLV, then it shall ignore that TLV.

8. IEEE 1905.1 topology discovery protocol

The 1905.1 topology discovery protocol enables a 1905.1 management entity to discover other 1905.1 devices and IEEE 802.1 bridges, and to populate a 1905.1 topology database. This protocol also enables notification of changes in network topology.

The 1905.1 topology discovery protocol enables a 1905.1 management entity to determine which devices are reachable by it and, if desired, to infer a more complete network topology. This protocol also enables the HLE to be notified by other 1905.1 devices as their topology changes.

The 1905.1 topology discovery protocol consists of a multicast discovery procedure, a unicast topology query/response procedure, and a relayed multicast topology notification procedure. The discovery procedure enables each 1905.1 management entity to discover the existence of its neighbors and to infer the presence or absence of one or more IEEE 802.1 bridges between it and a neighboring 1905.1 device. The topology query/response procedure enables a 1905.1 management entity to obtain information about another 1905.1 device as well as that device's neighbors. The topology notification procedure enables a 1905.1 management entity to be notified that the topology at another 1905.1 device has changed.

8.1 IEEE 802.1 bridge discovery

Each 1905.1 device sends two types of multicast discovery messages on each of its interfaces: an IEEE 802.1 bridge discovery message and a topology discovery message. An IEEE 802.1 bridge discovery message is an LLDPDU sent to the LLDP nearest bridge multicast address (01-80-C2-00-00-0E), which is not forwarded by the IEEE 802.1 bridge. The 1905.1 abstraction layer does not affect LLDP normal operations. A topology discovery message is sent to the 1905.1 multicast MAC address, which is forwarded by an IEEE 802.1 bridge but not by a 1905.1 device.

The 1905.1 topology discovery messages transmitted from the same 1905.1 interface as LLDPDUs should use the same MAC address in their MAC address TLV (Table 6-8) as the MAC address of the LLDPDU PortID TLV (8.5.3.3 of IEEE Std 802.1AB-2009).

These two messages allow a 1905.1 management entity to infer if another 1905.1 management entity is connected through one or more IEEE 802.1 bridges on a particular interface, as follows:

— No IEEE 802.1 bridge detected: A 1905.1 management entity receives, over a particular one of its 1905.1 device's 1905.1 interfaces, both nearest bridge group addressed advertisements and 1905.1 MAC group addressed advertisements from the same 1905.1 interface of the neighbor 1905.1 management entity.

  In this case, the Boolean flag "IEEE802.1BridgeFlag" is set to FALSE for this particular interface.

— Linked through one or more IEEE 802.1 bridges: A 1905.1 management entity either:

  1) Receives *only* 1905.1 MAC group addressed advertisements from the neighbor 1905.1 management entity on that particular interface.

  2) On a particular interface, receives both LLDPDUs and 1905.1 topology discovery messages with different MAC addresses in the 1905.1 topology discovery message MAC address TLV (Table 6-8) and in the LLDPDU PortID TLV (8.5.3.3 of IEEE Std 802.1AB-2009).

     In this case, the Boolean flag "IEEE802.1BridgeFlag" is set to TRUE for this particular interface.

This scheme does not enable a 1905.1 management entity to discover bridges that are not compliant with IEEE Std 802.1D-2004 (i.e., hubs where any address is transparently forwarded on all ports).

A 1905.1 management entity may construct a more complete network map by sending topology query messages to each neighboring 1905.1 management entity to obtain that device's neighbors (via topology response messages). A neighboring device is a 1905.1 device from which the multicast topology discovery message was received. A 1905.1 management entity may query any other 1905.1 device, e.g., by sending the topology query message to the neighbor's neighbors, and so forth, to the extent desired.

8.1.1 Transmission process

LLDPDUs constructed by the 1905.1 management entity shall comply with the LLDPDU format specified by 8.5 of IEEE Std 802.1AB-2009 and the format specified in 6.1 of this standard.

The 1905.1 abstraction layer shall not modify LLDPDUs constructed by an LLDP entity.

8.1.2 Reception process

The 1905.1 abstraction layer shall not prevent the LLDPDU to be delivered to the LLDP entity.

The 1905.1 abstraction layer shall ignore LLDP TLVs not specified in 6.1.

8.2 Topology discovery protocol

The topology discovery protocol uses the multicast discovery procedure, the unicast topology query/response procedure, and the relayed multicast topology notification procedure to enable a 1905.1 management entity to discover the network topology. These procedures make use of the following messages:

— Topology discovery message (neighbor multicast)
— IEEE 802.1 bridge discovery message (neighbor multicast)
— Topology query message (unicast)
— Topology response message (unicast)
— Topology notification message (relayed multicast)

8.2.1 Multicast discovery procedure

A 1905.1 management entity discovers the existence of other 1905.1 devices and IEEE 802.1 bridges (to the extent possible), through the use of two multicast discovery messages: the topology discovery message and the IEEE 802.1 bridge discovery message, described as follows.

8.2.1.1 Topology discovery message (neighbor multicast)

If the following event occurs, then within 1 s, a 1905.1 management entity shall transmit a topology discovery message as described in 7.2 and the topology discovery message format as described in 6.3.1:

— 60 s have elapsed since the last topology discovery message was sent.

If an implementation-specific event occurs (e.g., device initialized or an interface is connected), then a 1905.1 management entity shall transmit a topology discovery message using procedures described in 7.2 and message formats described in 6.3.1.

A 1905.1 management entity may wait for an implementation-specific event (e.g., all interfaces are ready after initialization) before sending its first topology discovery message.

The topology discovery message shall contain a new MID value, per 7.8.

8.2.1.2 IEEE 802.1 bridge discovery message (neighbor multicast)

If a 1905.1 management entity transmits a topology discovery message, then it shall also transmit an IEEE 802.1 bridge discovery message as described in 7.2 and the IEEE 802.1 bridge discovery message format as described in 6.1.

8.2.2 Topology query/response procedure

The topology query/response procedure enables a 1905.1 management entity to query and receive topology information from another 1905.1 management entity through the use of two unicast messages: the topology query message and the topology response message, described in 8.2.2.1 and 8.2.2.2.

8.2.2.1 Topology query message (unicast)

A 1905.1 management entity may send a topology query message to another 1905.1 management entity as described in 7.4 and the format of a topology query message as described in 6.3.2.

Each topology query message shall contain a new MID value, per 7.8.

8.2.2.2 Topology response message (unicast)

If a 1905.1 management entity receives a topology query message, then within 1 s it shall respond with a topology response message as described in 7.4 and the format of a topology response message as described in 6.3.3.

The topology response message shall contain the same MID that was received in the topology query message.

8.2.2.3 Topology notification message (relayed multicast)

If a 1905.1 management entity detects that any of the information specified to be sent in a topology response message has changed, then it shall within 1 s:

— Construct a topology notification message containing a new MID value per 7.8
— Propagate the message as described in 7.2 and the format of a topology notification message as described in 6.3.4

If a 1905.1 management entity receives a topology notification message, then it shall propagate it as described in 7.6.

9. IEEE 1905.1 security setup

9.1 Framework

This standard defines a unified security setup aimed to allow a 1905.1 device to join a given 1905.1 network through a single action for all the 1905.1 underlying network technologies supported by the device. A 1905.1 interface is considered authenticated when the underlying networking technology encryption mode has been successfully configured.

This standard defines three security setup methods for 1905.1 networks as follows:

a) 1905.1 push button configuration (1905.1 PBC) method, in which the user pushes a button on one 1905.1 device and then pushes a button on another 1905.1 device within a certain amount of time to cause these two devices to join the same network over a secured link encrypted with a shared key.

b) 1905.1 user configured passphrase/key (1905.1 UCPK) method, in which the 1905.1 network key is locally configured on each 1905.1 device.

c) 1905.1 near-field communication network key (1905.1 NFCNK) method, in which the 1905.1 devices are equipped with an NFC interface. The user touches the NFC interface with a second NFC device (e.g., a smartphone or a TV remote control) enabled to carry the 1905.1 network key. When a second or subsequent 1905.1 device is touched using the key-carrying device (KCD), the key is forwarded to the second or subsequent 1905.1 device.

   The 1905.1 NFCNK method is convenient for the user especially when multiple devices mounted on various fixed locations in a home shall be configured using an identical key (e.g., a satellite dish including a receiver, a high-definition television, a wireless router, a baby monitor or door camera, a personal computer, an audio system, and an audio video home server are connected via a power line and coaxial network). An NFC smartphone as KCD forwards the 1905.1 network key to the multiple devices. The user simply walks to the location where each device is mounted and touches the NFC interface using a smartphone. The security of this method is very high because the key is carried on a second independent medium. Unlike the push button method, there is no certain time limit until when the user has to touch the remotely installed devices.

Any given 1905.1 network may use either 1905.1 PBC or 1905.1 UCPK. In addition, the 1905.1 network may use NFCNK.

9.2 1905.1 security setup methods

A 1905.1 device shall support the 1905.1 PBC method. A 1905.1 device should support the 1905.1 NFCNK method and/or the 1905.1 UCPK method.

9.2.1 1905.1 UCPK setup method

In the 1905.1 UCPK setup method, the same single 256-bit 1905.1 network key is locally configured on each 1905.1 device for all 1905.1 interfaces. The user shall be able to configure the 1905.1 network key as a sequence of 64 hex characters.

A user may choose to configure a 1905.1 network passphrase as an alternative to configuring a 1905.1 network key. The 1905.1 network passphrase shall be a sequence of between 8 and 63 US-ASCII-encoded characters in the range of US-ASCII [0x20] to US-ASCII [0x7E].

9.2.1.1 U-key derivation from 1905.1 network key

If a 1905.1 network key is configured through the 1905.1 UCPK method, then the 1905.1 device shall derive the 1905.1 interface underlying network technology u-keys, by:

a) Computing the hash digest through the HMAC-SHA-256 function per section 8.3 of RFC-6234 that references FIPS180-2 to produce a 256-bit hash digest using the following parameters:
   1) whichSha: [in] = SHA256
   2) message_array[ ]: [in] = (see Table 9-1)
   3) length: [in] = the length of the message in message_array
   4) key[ ]: [in] = 1905.1 network key
   5) key_len: [in] = the length of the 1905.1 network key 256 bits
   6) digest[ ]: [out] = the 256-bit digest to be returned

Table 9-1—InterfaceType message_array

Interface type	Message_array
IEEE 802.11	“1905 easily creates interoperable Hybrid networks with deployed Wi-Fi”
IEEE 1901	“1905 easily creates interoperable Hybrid networks with deployed 1901”
MoCA	“1905 easily creates interoperable Hybrid networks with deployed MoCA”
Ethernet	“1905 easily creates interoperable Hybrid networks with deployed Ethernet”

b) Filtering the resulting hash digest as described in 9.2.1.1.1 through 9.2.1.1.4 to generate the 1905.1 interface underlying network technology-specific u-key. Note that the u-key generated for Ethernet is not currently used because Ethernet does not offer encryption.

9.2.1.1.1 WSC PSK

The WPA/WPA2 passphrase u-key (n characters) is the least significant $4 \times n$-bit hash digest expressed in hexadecimal using lowercase ASCII characters, where n is defined as follows:

— If the 1905.1 network key is locally configured, then n is equal to 62.
— Otherwise (the user chooses to configure a 1905.1 network passphrase):
   1) If the length of the 1905.1 passphrase is between 8 and 31 inclusive, then n is equal to two times the length of the 1905.1 passphrase.
   2) If the length of the 1905.1 passphrase is 32 or more, then n is equal to 62.

9.2.1.1.2 IEEE 1901 in-home shared key DSNA NMK (network key NMK-HS)

The IEEE 1901 in-home shared key DSNA NMK (direct entry NMK-HS) u-key is the least significant 128 bits of the hash digest (see 7.10.1.2.2 of IEEE Std 1901-2010).

9.2.1.1.3 IEEE 1901 PSNA pairwise key (PWK)

The IEEE 1901 PSNA PWK u-key is the least significant 128 bits of the hash digest (see 7.11.2.2 of IEEE Std 1901-2010). The 1905.1 management entity uses the "upper layer connection" method of sharing the PWK (see 7.11.3 of IEEE Std 1901-2010).

9.2.1.1.4 MoCA 1.1 privacy password

The MoCA 1.1 privacy password u-key is 17 decimal digits long including leading zeros derived from the 68 least significant bits of the hash digest with each digit derived from the modulo 10-decimal of the hexadecimal value from each 4 bits of the hash digest (see 6.3 of MoCA 1.1).

9.2.1.2 1905.1 network key derivation from 1905.1 network passphrase

The 1905.1 network passphrase mapping uses the PBKDF2 method from PKCS #5 to derive the 1905.1 network key (PNK) using the following parameters:

- PNK = PBKDF2 (1905.1NetworkPassphrase, Salt, 4096, 256)
- 1905.1NetworkPassphrase = 1905.1 network passphrase
- Salt = 1905.1 network name
- 4096 = the number of times 1905.1NetworkPassphrase is hashed
- 256 = the number of bits of PNK

The 1905.1 network name and 1905.1 network passphrase are user inputs. Both shall be a sequence of ASCII-encoded characters in the range of ASCII [0x20] to ASCII [0x7E].

The length of the 1905.1 network passphrase is between 8 and 63 characters.

The length of the 1905.1 network name is between 1 and 63 characters.

9.2.2 1905.1 PBC setup method

The 1905.1 PBC method works between two 1905.1 devices on the same 1905.1 network.

An example of the 1905.1 PBC method is illustrated in Figure 9-1.

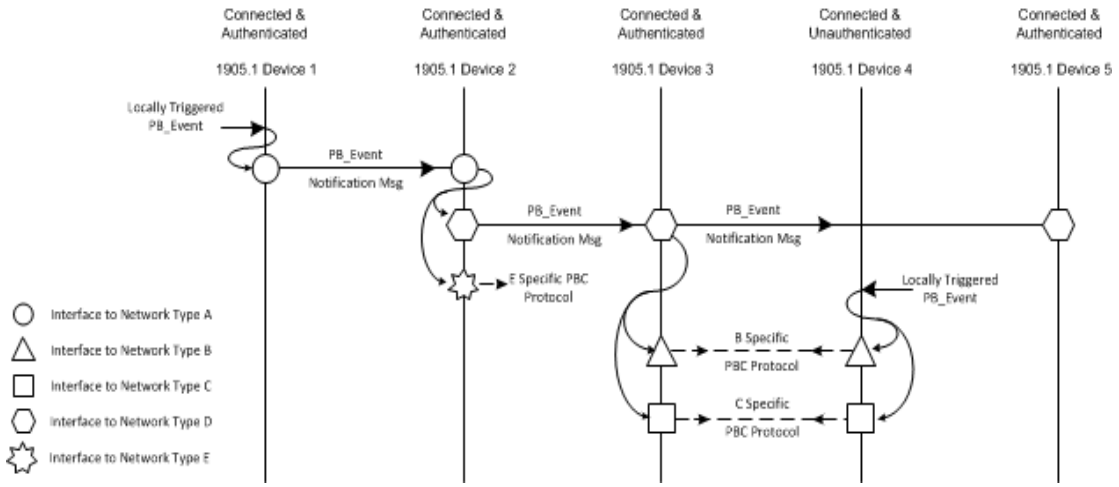

Figure 9-1—Example of 1905.1 push button event notification and 1905.1 push button configuration

9.2.2.1 1905.1 push button event handling

If the physical or logical PBC button is pushed on a 1905.1 device and if an underlying network technology-specific push button configuration sequence is not currently being performed on any of the network interface of this 1905.1 device, then a push button event is triggered on this 1905.1 device.

If a push button event is triggered on a 1905.1 device, then the 1905.1 management entity shall:

— For each interface, if the underlying interface is either:
  1) A non-IEEE 802.11,
  2) An IEEE 802.11 AP, and
  3) An IEEE 802.11 STA (non-AP) that is not associated with any AP,

  then initialize the underlying network technology-specific push button configuration sequence on the 1905.1 interfaces supporting push button configuration methods.

— Send a 1905.1 push button event notification message (see 6.3.11) as described in 7.3. In the Push_Button_Event notification TLV of the 1905.1 push button notification message, include the media type and media-specific information for all interfaces for which the push button configuration was initialized, if any.

9.2.2.2 1905.1 push button event notification handling

Figure 9-2 describes how a 1905.1 device handles a 1905.1 push button event notification message.

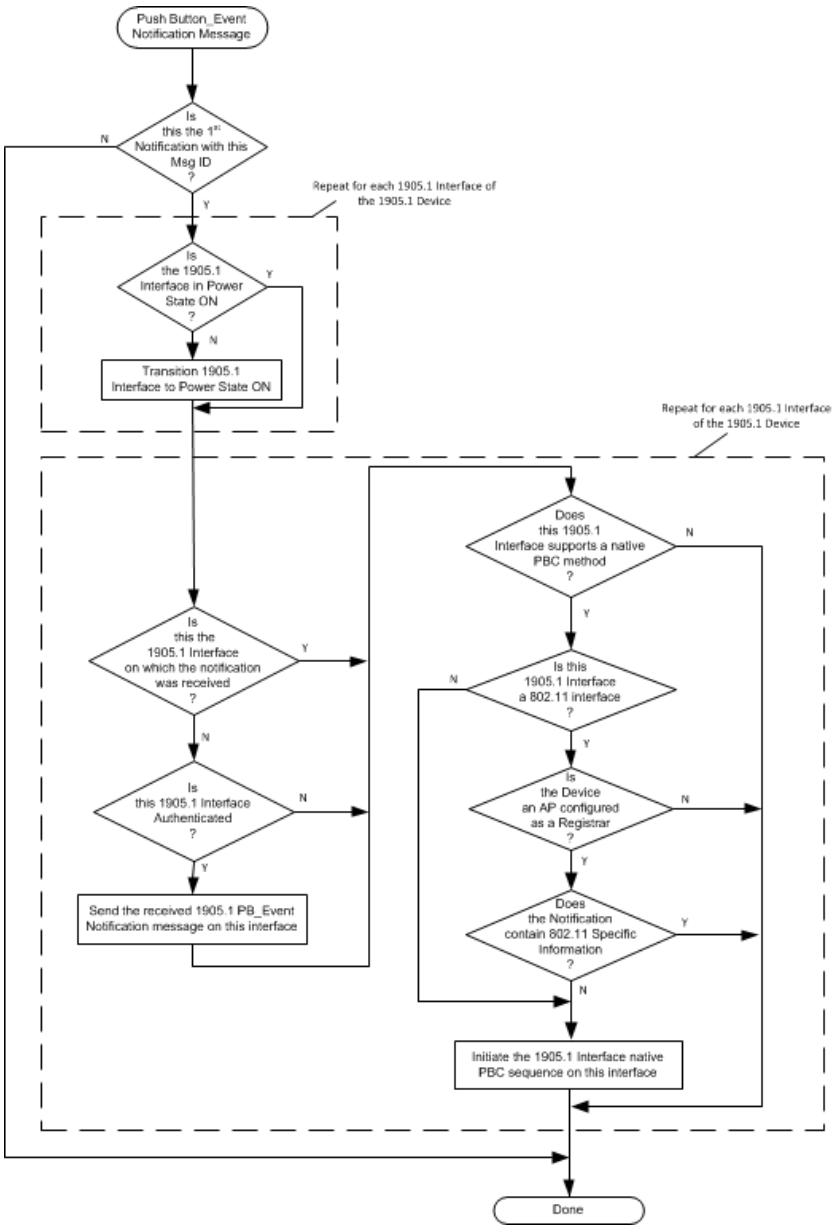

Figure 9-2—Push button event notification handling

If a 1905.1 management entity receives a 1905.1 push button notification message (see 6.3.11), then it shall:

a) Relay the received 1905.1 push button notification message as described in 7.6.

b) Initialize the underlying network technology-specific push button configuration sequence on all non-IEEE 802.11 1905.1 interfaces supporting push button configuration methods.

c) If the underlying network technology is an IEEE 802.11 AP and the device is configured as the registrar, and the push button event notification does not contain IEEE 802.11 media type information, then initialize the underlying push button configuration sequence on the IEEE 802.11 interface(s).

9.2.2.3 1905.1 push button join notification generation and handling

Upon successful completion of a push button operation that results in the joining of a new device to the 1905.1 network, the 1905.1 management entity of the 1905.1 device(s) that initiated the push button configuration sequence shall:

— Send a 1905.1 push button join notification message (see 6.3.12) as described in 7.3.

If two or more push button join notification messages are received within an implementation defined time interval with all of the following conditions:

a) The MID fields in the push button join notification TLVs have the same value.

b) The AL ID fields in the push button join notification TLVs have the same value.

c) The AL ID fields in the 1905.1 AL MAC address type TLVs have different values.

Then the 1905.1 management entity of the 1905.1 devices may send a notification to the upper layers.

9.2.3 1905.1 NFCNK setup method

A KCD is a device that shall be able to:

— Store a 1905.1 network key
— Generate a 1905.1 network key
— Display a 1905.1 network key to the user

With NFCNK, the following use cases may apply:

a) If a 1905.1 network key is locally configured on the 1905.1 device and the NFC KCD does not contain a 1905.1 network key, then the KCD may copy the 1905.1 network key from the 1905.1 device.

b) If an NFC KCD contains a 1905.1 network key and a 1905.1 device does not have a 1905.1 network key, then the 1905.1 device may copy the 1905.1 network key from the KCD.

c) If an NFC KCD contains a 1905.1 network key and a 1905.1 device is locally configured with a different 1905.1 network key, then the higher layer entity may be notified.

d) If an NFC KCD does not contain a 1905.1 network key and a 1905.1 device is not locally configured with a 1905.1 network key, then the higher layer entity may be notified.

Furthermore, before copying the 1905.1 network key to a 1905.1 device, validation of the key may be necessary (e.g., NFC may provide a secured way to transfer or copy the 1905.1 device key and the higher layer entities of KCD and the 1905.1 device to authenticate each other).

Figure 9-3 shows a block schematic of the portable device carrying the key (KCD) and several 1905.1 devices equipped with several network PHY layers.

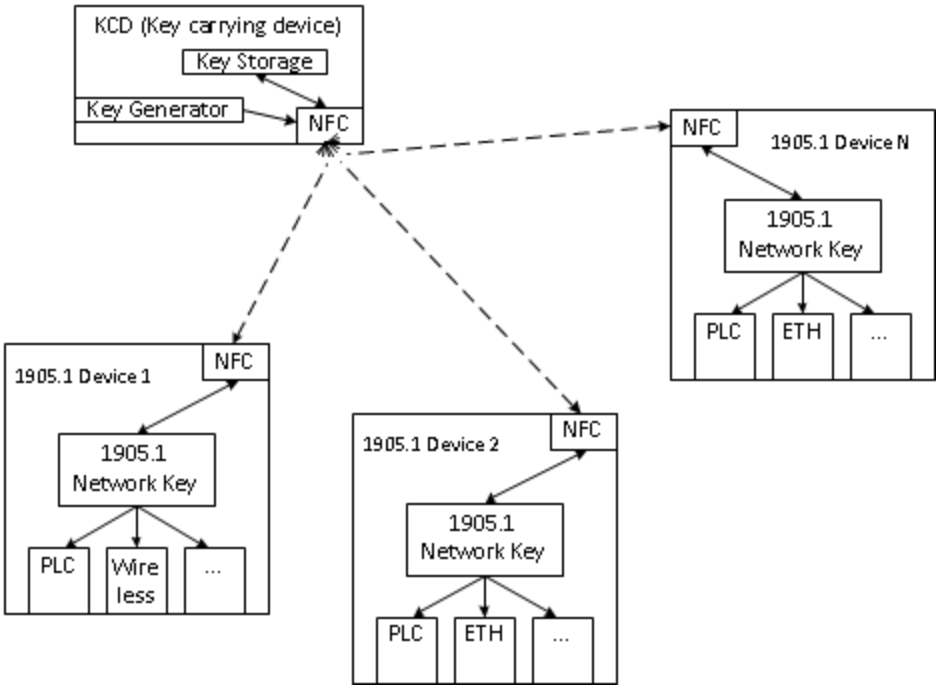

Figure 9-3—NFCNK device overview

The higher layer entity on the KCD may provide a selection possibility to the user if the network key to be used is generated new, taken from the 1905.1 device or taken from the key storage. If the user likes to set up a new 1905.1 network, the initial network key has to be generated. To add further devices to the 1905.1 network, this network key is stored and used when additional devices are added.

9.2.3.1 NFC message exchange procedure

If a KCD is touched to a 1905.1 device, both devices shall exchange messages according to the NFC Forum Connection Handover Technical Specification for a negotiated handover.

For example, the KCD may send a Handover Request Message indicating 1905.1 capability by using

vnd.ieee.1905.nfcnk

as the carrier type name in a handover carrier record.

9.2.3.2 1905.1 network key record

The 1905.1 NFC configuration (NFCC) request record type name shall be the MIME media type vnd.ieee.1905.nfcnk and shall contain the following payload shown in Table 9-2.

Table 9-2—Near-field communication data exchange format (NDEF) record payload of NFCC network key record

Field	R/O/C	Length	Value
Type	R	1 Byte	"0x01"
1905.1 network key	R	32 octets	The 1905.1 network key

10. Protocols for IEEE 802.11 access point autoconfiguration with IEEE Std 1905.1

10.1 Operation of AP-autoconfiguration

The AP-autoconfiguration process uses CMDUs to convey IEEE 802.11 parameters from a registrar to an AP enrollee to set up the initial configuration or renew an existing configuration of an IEEE 802.11 interface. This operation provides an automatic method to set up an extended service set of multiple APs and to keep them synchronized. An IEEE 802.11 AP with an unconfigured IEEE 802.11 interface acts as an AP enrollee and uses an authenticated 1905.1 interface to reach another 1905.1 device acting as a registrar.

The 1905.1 network shall be configured with a single registrar for the 1905.1 push button configuration (see 9.2.2) to work properly on 1905.1 devices that contain IEEE 802.11 interfaces.

The successful authentication of a 1905.1 interface triggers the AP-autoconfiguration process. The AP-autoconfiguration process operates in the following two phases:

a) Registrar discovery phase to get information about the registrar available on the 1905.1 network.
b) IEEE 802.11 parameter configuration phase: To transfer ConfigData (as specified in Wi-Fi simple configuration) between the registrar and the AP enrollee. The 1905.1 abstraction layer provides a transparent transport protocol for WSC frames M1 and M2.

After the registrar receives M1, it sends the ConfigData encrypted by the KeyWrapKey (as specified in Wi-Fi simple configuration) in M2.

This process is defined per interface and shall be repeated for each of the unconfigured IEEE 802.11 AP interfaces.

10.1.1 Registrar discovery phase

If the 1905.1 device contains an unconfigured IEEE 802.11 AP interface, then the registrar discovery phase shall start after successful authentication of a 1905.1 interface.

The AP enrollee sends an AP-autoconfiguration search message to discover the registrar. This multicast search message includes the UnconfiguredFreqBand TLV to discover whether the registrar is capable of supporting the autoconfiguration for the requested frequency band. If the registrar supports the requested capabilities, then it sends back a unicast AP-autoconfiguration response message.

The AP-autoconfiguration search message shall contain a new MID value, per 7.8.

The AP-autoconfiguration response message shall contain the same MID that was received in the AP-autoconfiguration search message.

Figure 10-1 illustrates 1905.1 CMDUs used for the registrar discovery and for the IEEE 802.11 parameter configuration of two unconfigured interfaces.

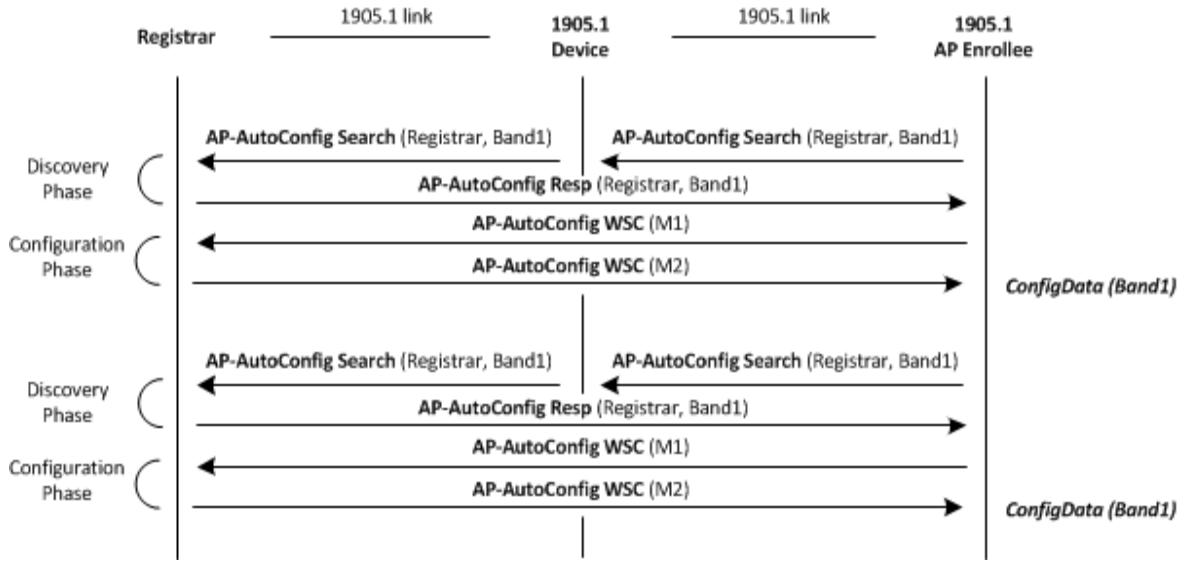

Figure 10-1—1905.1 CMDU exchange for initial setup of an AP with two unconfigured interfaces

10.1.2 IEEE 802.11 parameter configuration phase

The IEEE 802.11 parameter configuration phase shall start after the reception of an AP-autoconfiguration response message.

The IEEE 802.11 parameter configuration phase is accomplished by exchanging messages: AP-autoconfiguration WSC (M1) and AP-autoconfiguration WSC (M2) (content and format of M1 and M2 as defined in Wi-Fi simple configuration). The detailed description of M1 and M2 frame content and format is outside the scope of this standard.

The IEEE 802.11 configuration data are delivered to the AP enrollee in M2. Table 10-1 shows a list of attributes available in this WSC message.

An AP-autoconfiguration WSC message shall contain a new MID value, per 7.8.

After the registrar receives M1, it shall send the ConfigData encrypted by the KeyWrapKey (as specified in Wi-Fi simple configuration) in M2.

If the IEEE 802.11 parameter configuration phase is not completed successfully, the AP enrollee shall restart the registrar discovery phase in 10.1.1.

Table 10-1—IEEE 802.11 settings (ConfigData) in M2 frame

Attribute	Description
SSID	
Authentication type	The authentication type to be used by the AP.
Encryption type	The encryption type to be used by the AP.
Network key index	This field shall be omitted.
Network key	
MAC address	AP's MAC address (BSSID).
New password	This field shall be omitted.
Device password ID	This field shall be omitted.
<other…>	Multiple attributes are permitted.
Key wrap authenticator	

The IEEE 802.11 parameter configuration phase is accomplished without any user input on the registrar because the connection between the AP enrollee and the registrar uses authenticated 1905.1 links.

10.1.3 Renewing of configuration

An AP-autoconfiguration renew message shall be multicast by the registrar to inform the AP enrollees configured with the AP-autoconfiguration process to restart the process to get updated values. This renewing of configuration is important to keep a synchronization of configuration parameters in the extended service set in the lifetime of the 1905.1 network (reboot, upgrade operations, etc.).

The AP-autoconfiguration renew message shall contain a new MID value, per 7.8.

All AP enrollees previously configured by a registrar shall start the IEEE 802.11 parameter configuration phase after the reception of an AP-autoconfiguration renew message.

This AP-autoconfiguration process is defined per frequency band and shall be repeated multiple times if the registrar supports configuration for multiple bands.

Figure 10-2 illustrates 1905.1 CMDUs used for the renewal of configurations when the registrar supports IEEE 802.11 settings for two frequency bands.

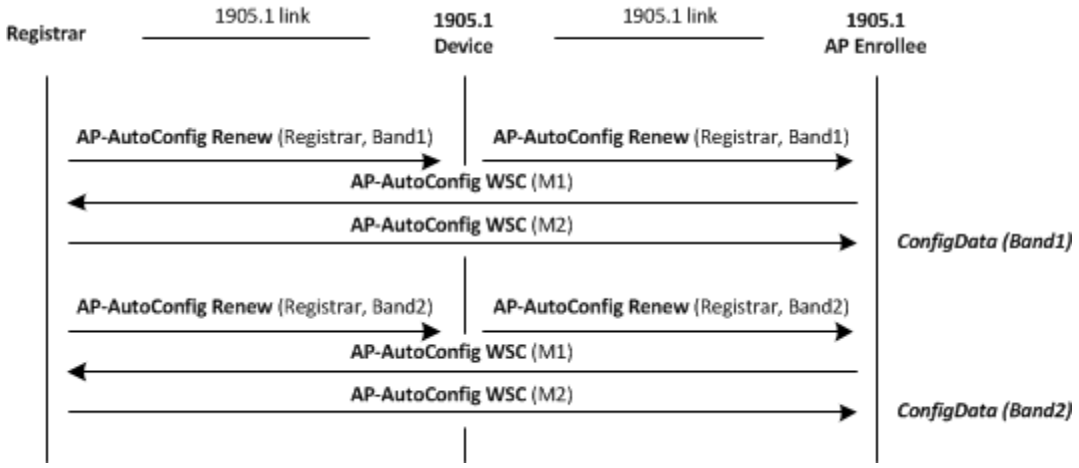

Figure 10-2—1905.1 CMDU exchange to renew configuration of an AP with two interfaces

11. Link metrics

A 1905.1 device provides link metrics information for each one of its authenticated 1905.1 interfaces through a 1905.1 ALME primitive that triggers 1905.1 link metric information dissemination protocol messages.

The 1905.1 link metrics parameters described in Table 6-18 and Table 6-20 are the parameters of the transmission channel of a 1905.1 link, which are specified per {1905.1 MAC abstraction layer address, 1905.1 intfAddress} tuple.

11.1 Link metric information dissemination protocol

The link metric information dissemination protocol enables a 1905.1 management entity to obtain link metric information at another 1905.1 device. The receiving 1905.1 device provides information regarding link metrics related to all of its interfaces to 1) a particular 1905.1 neighbor device or 2) all of its 1905.1 neighbor devices.

The 1905.1 link metric information dissemination protocol consists of a procedure with link metric query message and link metric response message.

11.1.1 Link metric query message (unicast)

A link metric query message may be sent from a 1905.1 management entity to another 1905.1 management entity, requesting the link metrics for all the interfaces between the receiving 1905.1 device (the one receiving the link metric query message) and either a specified 1905.1 neighbor device or all its 1905.1 neighbor devices.

The link metric query message shall be sent as a CMDU unicast transmission (7.4) with the Link metric query message format described in 6.3.5. The link metric query message shall contain a new MID value described in 7.8.

11.1.2 Link metric response message (unicast)

If a 1905.1 management entity receives a link metric query message, then it shall respond with a link metric response message as described in 7.4 and the format as described in 6.3.6. The link metric response message shall contain the same MID that was received in the link metric query message.

Annex A

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] IEEE Standards Association, FAQs: The registration authority. Web page. Organizationally unique identifiers (OUIs), as per http://standards.ieee.org/faqs/OUI.html.

[B2] ISO/IEC 8802-2-1998, Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 2: Logical Link Control.[19]

[B3] Wi-Fi Peer-to-Peer (P2P) Specification v1.1; Wi-Fi Alliance.[20]

[19] ISO/IEC publications are available from the ISO Central Secretariat (http://www.iso.org/). ISO publications are also available in the United States from the American National Standards Institute (http://www.ansi.org/).
[20] Wi-Fi Alliance publications are available from the Wi-Fi Alliance (http://www.wi-fi.org/).

Annex B

(normative)

UCPK test vectors

In this annex, six test vectors are provided for the 1905.1 network key as well as for the underlying keys and passwords. The input is a user 1905.1 network passphrase.

Test vector 1
Input
1905.1 Network Passphrase = password
1905.1 Salt = Backward interoperability is a feature of 1905.1

Output
1905.1 NK = 4320B5D140946AB78CC78B081BA0FBE0402E8C1D5FD9763B993825DB38896678
WiFi passphrase = 678492b577bc00ee5de390a754ff734dcf178bca76179dbd50c9e1c7e9be38
1901 NMK/PWK = aa440e5ccfe785b15f075a690904006b
MoCA password = 01152751561285611

Test vector 2
Input
1905.1 Network Passphrase = my secret
1905.1 Salt = Backward interoperability is a feature of 1905.1

Output
1905.1 NK = 5625B2CB4E4B5EB9A1A6DF8132344C48B432A1BAAB8B3CF7DA8787FB82686E65
WiFi passphrase = 42780c92da6706ddd81e247c3faa68b801ae84d6f37e33654aa4dfbfcea56d
1901 NMK/PWK = 12068150d9dcfcc54b54adccebde42d1
MoCA password = 06341151528707840

Test vector 3
Input
1905.1 Network Passphrase = My passphrase is huge, its length is 44 char
1905.1 Salt = Backward interoperability is a feature of 1905.1

Output
1905.1 NK = 78B92C06C3FC7588CC2E460CDB8A21BBE4491EB395360F0EF1D60A893CC47BAC
WiFi passphrase = a72c5a8c05cb806cb22120f0b4877fda9fa119ba89ff7f5987199ac804c132
1901 NMK/PWK = 6f2f3a0066ca74c2bbe340c346054376
MoCA password = 74600109479533005

Test vector 4
Input
1905.1 Network Passphrase = password
1905.1 Salt = My P1905 Network Name
Output
1905.1 NK = 1D1719F80CCEBF35468697C7B9F158F57EE45F5632F12DEA59529001945164D7
WiFi passphrase = 86cbc41b9d3fcc1c96d15485f1193243e42f731ef386b2bc4a5abe87998982
1901 NMK/PWK = 81fdc883614d72ee417f01a011d50fa8
MoCA password = 30225578379007175

Test vector 5
Input
1905.1 Network Passphrase = my secret
1905.1 Salt = My P1905 Network Name
Output
1905.1 NK = 25D98D0128F4E5F9F2DA31FD13EC89358F81D1130C82BC2B8EF11795476B11EC
WiFi passphrase = 68daaedfe47b6ca279d696ca1b282611df43f5dbc49cda997c5c9077203c3d
1901 NMK/PWK = 1e1bb24964579c019904d9aa3acfaa27
MoCA password = 50232104532353603

Test vector 6
Input
1905.1 Network Passphrase = My passphrase is huge, its length is 44 char
1905.1 Salt = My P1905 Network Name
Output
1905.1 NK = 0E6765FF5072D6239283A800AD29B7D9B38219029AB07C68DB2FE68847992C2A
WiFi passphrase = 6c034fd3e0d306bb9aa5d0e69b2ef72d04024c17ec449cfc14a72f37ee82f3
1901 NMK/PWK = 0802ba8c185e5369471210ccc8ba8087
MoCA password = 20403041352511556

Annex C

(informative)

IEEE 1905.1 data models

The 1905.1 data models use the Device:2 data model for the CPE WAN management protocol (TR-069).[21]

C.1 Introduction

This annex describes the 1905.1 device data model for the CPE WAN management protocol (CWMP). TR-069 defines the generic requirements of the management protocol methods that can be applied to any TR-069 CPE. The 1905.1 data model is in addition to any other data models supported by a 1905.1 device (e.g., TR-098 and TR-181 i2 data models). Figure C-1 shows the hierarchy of the data model object in the Device:2 data model structure.

The 1905.1 device data model follows the conventions defined in TR-106 for versioning of data models and the use of profiles.

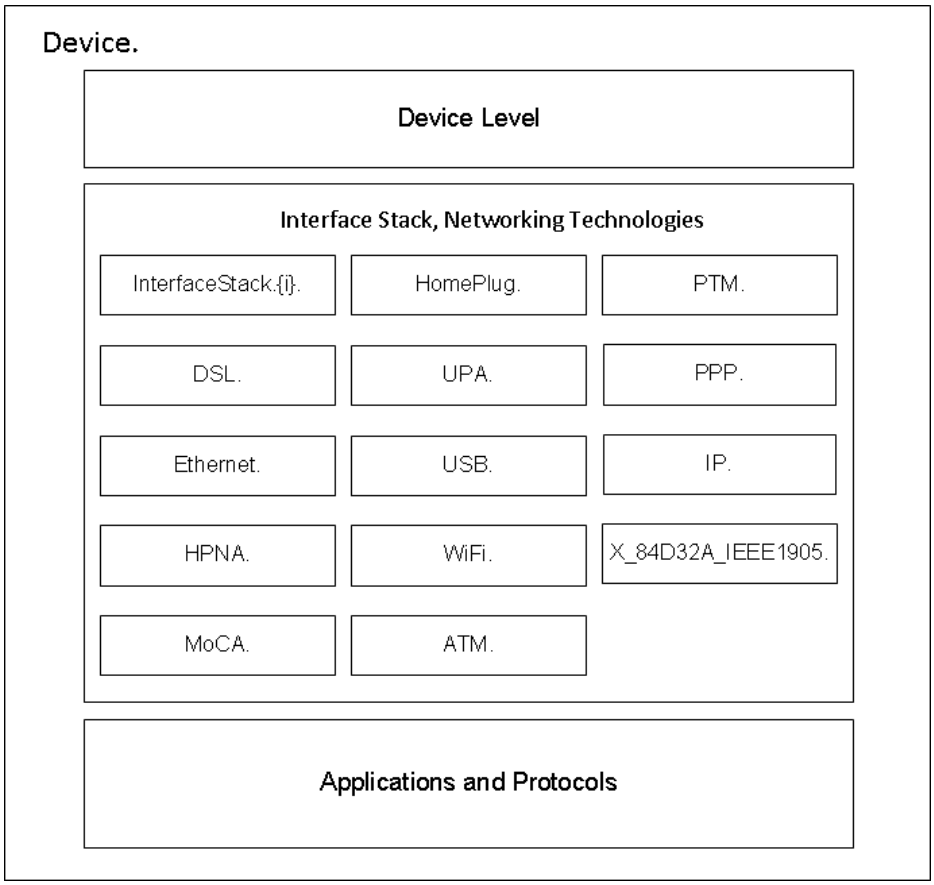

Figure C-1—Hierarchy of the data model object in the Device:2 data model structure

[21] The final data model will be addressed by the BBF.

C.2 Architecture

As required by TR-181 i2, every interface (protocol) object in the 1905.1 device must specify a LowerLayers parameter in its data model, which is a list of interface objects that are stacked immediately below it. The data model for a higher layer protocol that has the 1905.1 AL layer stacked immediately below it (e.g., LLC) must include the 1905.1 object in its LowerLayers parameter list. Similarly, a 1905.1 object must include the underlying network technology interfaces in its LowerLayers parameter list. Given a set of interfaces and their LowerLayers parameters, the CPE can autogenerate an InterfaceStackTable, which provides a representation of the entire networking stack in the device.

C.3 Data model definition

C.3.1 General notation

Parameter names use a hierarchical form similar to a directory tree. The name of a particular parameter is represented by the concatenation of each successive node in the hierarchy separated with a "." (dot), starting at the trunk of the hierarchy and leading to the leaves. When specifying a partial path, indicating an intermediate node in the hierarchy, the trailing "." (dot) is always used as the last character.

Parameter names shall be treated as case sensitive.

In some cases, where multiple instances of an object can occur, the placeholder node name "{i}" is shown. In actual use, this placeholder is to be replaced by an instance number, which must be a positive integer (≥1). In some cases, object instances may be deleted; therefore, instance numbers in general may not be contiguous.

All MAC addresses are represented as strings of 12 hexadecimal digits (digits 0 to 9, letters A through F, or letters a through f) displayed as six pairs of digits separated by colons. Unspecified or inapplicable MAC addresses *must* be represented as empty strings unless otherwise specified by the parameter definition.

C.3.2 Data types

The parameters defined in this specification make use of a limited subset of the default SOAP data types (SOAP1.1) and are shown in Table C-1. For the descriptions of data types, see TR-106a6.

Table C-1—Data types

Data Type	Base Type
Alias	string(64)
Dbm1000	int
IPAddress	string(45)
IPPrefix	string(49)
IPv4Address	IPAddress(15)
IPv4Prefix	IPPrefix(18)
IPv6Address	IPAddress
IPv6Prefix	IPPrefix
MACAddress	string(17)
StatsCounter32	unsignedInt
StatsCounter64	unsignedLong

C.4 X_84D32A_Device:2.5 data model

Table C-2 defines the X_84D32A_Device:2.5 data model. The table lists the objects defined for a 1905.1 device and the corresponding parameters within those objects. The version number associated with each object and parameter is shown in the "Version" column of Table C-2. For a given implementation of this data model, the device must indicate support for the highest version number of any object or parameter that it supports.

Name

The full name of a parameter is the concatenation of the object name shown in the yellow header with the individual parameter name.

Write

"W" indicates the parameter may be writable (if "W" is not present, the parameter is defined as read only). For an object, "W" indicates object instances can be added or deleted.

Object Default

The default value of the parameter on creation of an object instance via TR-069. A hyphen indicates that no default value is specified. For a parameter in which no default value is specified, on creation of a parent object instance, the 1905.1 must set the parameter to a value that is valid according to the definition of that parameter.

Version

The "Version" column indicates the minimum data model version required to support the associated parameter or object.

The following objects are applicable only when the optional forwarding entity is present:

— Device.X_84D32A_IEEE1905.AL.ForwardingTable
— Device.X_84D32A_IEEE1905.AL.ForwardingTable.ForwardingRule

Table C-2—X_84D32A_Device:2.5 data model

Name	Type	Write	Description	Object Default	Version
Device.	object	-	The top-level object for a Device.	-	2.0
Device.X_84D32A_IEEE1905.	object	-	This object represents the management functions for the 1905.1 capabilities as defined in [IEEE1905.1].	-	2.5
Version	unsignedInt	-	1905.1 Version Number.	-	2.5
Device.X_84D32A_IEEE1905.AL.	object	-	This object represents the management functions for the 1905.1 Abstraction Layer as defined in [sub-clause 4.4 Abstraction Layer/IEEE1905.1].	-	2.5
IEEE1905Id	string(17)	-	[*MACAddress*] 1905.1 AL MAC Address.	-	2.5
Status	string	-	The current operational state of the 1905.1 Abstraction Layer interface to upper interface layers (see [Section 4.2.2/TR-181i2]). Enumeration of: • *Up* • *Down* • *Unknown* • *Dormant* • *NotPresent*	-	2.5

Name	Type	Write	Description	Object Default	Version
			• *LowerLayerDown* • *Error* (OPTIONAL) It SHOULD change to *Up* if and only if the 1905.1 Abstraction Layer is able to transmit and receive network traffic; it SHOULD normally be *Down* when the interface cannot transmit and receive network traffic; it SHOULD change to *Dormant* if and only if the interface is operable but is waiting for external actions before it can transmit and receive network traffic (and subsequently change to *Up* if still operable when the expected actions have completed); it SHOULD change to *LowerLayerDown* if and only if the interface is prevented from entering the *Up* state because one or more of the interfaces beneath it is down; it SHOULD remain in the *Error* state if there is an error or other fault condition detected on the interface; it SHOULD remain in the *NotPresent* state if the interface has missing (typically hardware) components; it SHOULD change to*Unknown* if the state of the interface can not be determined for some reason. This parameter is based on *ifOperStatus* from [RFC2863].		
LastChange	unsignedInt	-	The accumulated time in *seconds* since the 1905.1 Abstraction Layer interface entered its current operational state.	-	2.5
LowerLayers	string(1024)	-	Comma-separated list (maximum length 1024) of strings. Each list item MUST be the path name of a row in the *AL.Interface* table. If the referenced object is deleted, the corresponding item MUST be removed from the list. See [Section 4.2.1/TR-181i2].	-	2.5
InterfaceNumberOfEntries	unsignedInt	-	The number of entries in the *Interface* table.	-	2.5
Device.X_84D32A_IEEE1905.AL.Interface.{i}.	object	-	The 1905.1 interface table (a stackable interface object as described in [sub-clause 5 Abstraction Layer Management/IEEE1905.1]). At most one entry in this table can exist with a given value for *InterfaceId*.	-	2.5
InterfaceId	string(17)	-	[*MACAddress*] MAC Address of this interface	-	2.5
Status	string	-	The current operational state of the interface (see [Section 4.2.2/TR-181i2]). Enumeration of: • *Up* • *Down* • *Unknown* • *Dormant* • *NotPresent* • *LowerLayerDown* • *Error* (OPTIONAL) It SHOULD change to *Up* if and only if the interface is able to transmit and receive network traffic; it SHOULD normally be *Down* when the interface cannot transmit and receive network	-	2.5

Name	Type	Write	Description	Object Default	Version
			traffic; it SHOULD change to *Dormant* if and only if the interface is operable but is waiting for external actions before it can transmit and receive network traffic (and subsequently change to *Up* if still operable when the expected actions have completed); it SHOULD change to *LowerLayerDown* if and only if the interface is prevented from entering the *Up* state because one or more of the interfaces beneath it is down; it SHOULD remain in the *Error* state if there is an error or other fault condition detected on the interface; it SHOULD remain in the *NotPresent* state if the interface has missing (typically hardware) components; it SHOULD change to *Unknown* if the state of the interface can not be determined for some reason. This parameter is based on *ifOperStatus* from [RFC2863].		
LastChange	unsignedInt	-	The accumulated time in *seconds* since the interface entered its current operational state.	-	2.5
LowerLayers	string(1024)	-	Comma-separated list (maximum length 1024) of strings. Each list item MUST be the path name of an interface object that is stacked immediately below this interface object. If the referenced object is deleted, the corresponding item MUST be removed from the list. See [Section 4.2.1/TR-181i2].	-	2.5
MediaType	string	-	Media type of this *Interface*. Enumeration of: • *IEEE 802.3u* (IEEE 802.3u Fast Ethernet) • *IEEE 802.3ab* (IEEE 802.3ab Gigabit Ethernet) • *IEEE 802.11b* (IEEE 802.11b (2.4GHz)) • *IEEE 802.11g* (IEEE 802.11g (2.4GHz)) • *IEEE 802.11a* (IEEE 802.11a (5GHz)) • *IEEE 802.11n 2.4* (IEEE 802.11n (2.4GHz)) • *IEEE 802.11n 5.0* (IEEE 802.11n (5GHz)) • *IEEE 802.11ac* (IEEE 802.11ac (5GHz)) • *IEEE 802.11ad* (IEEE 802.11ad (60GHz)) • IEEE 802.11af (IEEE 802.11af) • *IEEE 1901 Wavelet* (IEEE 1901 Wavelet) • *IEEE 1901 FFT* (IEEE 1901 FFT) • *MoCAv1.1* (MoCAv1.1)	-	2.5
PowerState	string	W	The Power State of this *Interface*. Enumeration of: • *On* • *Power_Save*	-	2.5

Name	Type	Write	Description	Object Default	Version
			• *Off* • *Unsupported*		
LinkNumberOfEntries	unsignedInt	-	The number of entries in the *Link* table.	-	2.5
Device.X_84D32A_IEEE1905.AL.Interface.{i}.Vendor Properties.	object	-	This object defines the vendor specific properties of this *Interface* as defined in [sub-clause 5.3.1 Vendor Specific Info IE/IEEE1905.1].	-	2.5
OUI	string(6:6)	-	Organizationally unique identifier of the manufacturer of this *Interface*. Represented as a six hexadecimal-digit value using all upper-case letters and including any leading zeros. Possible patterns: — *[0-9A-F]{6}* The value MUST be a valid OUI.	-	2.5
Information	hexBinary-(65535)	-	A hexbinary string used to to provide vendor specific information about this *Interface*.	-	2.5
Device.X_84D32A_IEEE1905.AL.Interface.{i}.Link.{i}.	object	-	This object defines the 1905.1 Link management properties]. At most one entry in this table can exist with the same values for *InterfaceId* and*IEEE1905Id*.	-	2.5
InterfaceId	string(17)	-	[*MACAddress*] MAC Address of the interface of the Neighbor for this *Link*.	-	2.5
IEEE1905Id	string(17)	-	[*MACAddress*] MAC Address of the 1905.1 AL entity of the Neighbor device on this*Link*.	-	2.5
MediaType	string	-	Media type of this *Link*. Enumeration of: • *IEEE 802.3u* (IEEE 802.3u Fast Ethernet) • *IEEE 802.3ab* (IEEE 802.3ab Gigabit Ethernet) • *IEEE 802.11b* (IEEE 802.11b (2.4GHz)) • *IEEE 802.11g* (IEEE 802.11g (2.4GHz)) • *IEEE 802.11a* (IEEE 802.11a (5GHz)) • *IEEE 802.11n 2.4* (IEEE 802.11n (2.4GHz)) • *IEEE 802.11n 5.0* (IEEE 802.11n (5GHz)) • *IEEE 802.11ac* (IEEE 802.11ac (5GHz)) • *IEEE 802.11ad* (IEEE 802.11ad (60GHz)) • IEEE 802.11af (IEEE 802.11af) • *IEEE 1901 Wavelet* (IEEE 1901 Wavelet) • *IEEE 1901 FFT* (IEEE 1901 FFT) • *MoCAv1.1* (MoCAv1.1)	-	2.5
Device.X_84D32A_IEEE1905.AL.Interface.{i}.Link.{i}.Metric.	object	-	This object represents the metrics for this *Link* as defined in [Tables 6-18, 6-	-	2.5

Name	Type	Write	Description	Object Default	Version
			20/IEEE1905.1].		
PacketErrors	unsignedInt	-	[*StatsCounter32*] Estimated number of lost *Packets* sent to the Neighbor on this *Link*during a measurement period.	-	2.5
TransmittedPackets	unsignedInt	-	[*StatsCounter32*] Estimated number of *Packets* sent to the Neighbor on this *Link*, in the same measurement period used to estimate *PacketErrors*.	-	2.5
PacketsReceived	unsignedInt	-	[*StatsCounter32*] Estimated number of *Packets* received from this Neighbor on this *Link*, in the same measurement period used to estimate *PacketErrors*.	-	2.5
MACThroughputCapacity	unsignedInt	-	The maximum MAC throughput in *Mb/s* between this *Interface* and the Neighbor on this*Link* that is estimated at this *Interface*.	-	2.5
LinkAvailability	unsignedInt-[0:100]	-	The estimated average *percent* of time that this *Link* is idle.	-	2.5
PHYRate	unsignedInt	-	The Physical Layer (PHY) rate in *Mb/s* between this *Interface* and the Neighbor on this*Link* that is estimated at this *Interface*.	-	2.5
RSSI	unsignedInt-[0:255]	-	The estimated Received Signal Strength Indicator (RSSI) ratio in *db* between this *Interface* and the Neighbor on this *Link* that is estimated at the receive side of this *Interface*. Note: This parameter is valid only for IEEE 802.11 Neighbors.	-	2.5
Device.X_84D32A_IEEE1905.AL.ForwardingTable.	object	-	This object represents the rules to forward PDUs between interfaces within the 1905.1 Abstraction Layer.	-	2.5
ForwardingRuleNumberOfEntries	unsignedInt	-	The number of entries in the *ForwardingRule* table.	-	2.5
Device.X_84D32A_IEEE1905.AL.ForwardingTable.ForwardingRule.{i}.	object	W	The 1905.1 fowarding rule as defined in [sub-clause 5 Abstraction Layer Management/IEEE1905.1].	-	2.5
InterfaceList	string(2048)	W	Result parameter indicating the list of interfaces to which a frame satisfying the following classification rule should be forwarded. Comma-separated list (maximum length 2048) of strings (maximum length 256). Each list item MUST be the path name of an object, which MUST be a row of an *Interface* object. If the referenced item is deleted, the corresponding item MUST be removed from the list.	<Empty>	2.5
MACDestinationAddress	string(17)	W	[*MACAddress*] Classification criterion. The destination MAC address. An empty string indicates this criterion is not used for classification.	-	2.5
MACDestinationAddressExclude	boolean	W	If *false*, the classification includes only those frames that match the*MACDestinationAddress* entry, if specified. If *true*, the classification includes all frames except those that match the*MACDestinationAddress* entry, if specified.	false	2.5
MACSourceAddress	string(17)	W	[*MACAddress*] Classification criterion.	-	2.5

Name	Type	Write	Description	Object Default	Version
			The source MAC address. An empty string indicates this criterion is not used for classification.		
MACSourceAddressExclude	boolean	W	If *false*, the classification includes only those frames that match the*MACSourceAddress* entry, if specified. If *true*, the classification includes all frames except those that match the*MACSourceAddress* entry, if specified.	false	2.5
EtherType	int[-1:]	W	Classification criterion. Ether Type Field in a frame. A value of -1 indicates this criterion is not used for classification.	-1	2.5
EtherTypeExclude	boolean	W	If *false*, the classification includes only those frames that match the *EtherType* entry, if specified. If *true*, the classification includes all frames except those that match the *EtherType*entry, if specified.	false	2.5
Vid	int[-1:]	W	Classification criterion. IEEE 802.1Q VLAN ID in a frame. A value of -1 indicates this criterion is not used for classification.	-1	2.5
VidExclude	boolean	W	If *false*, the classification includes only those frames that match the *Vid* entry, if specified. If *true*, the classification includes all frames except those that match the *Vid* entry, if specified.	false	2.5
PCP	int[-1:7]	W	Classification criterion. IEEE 802.1Q Priority Code Point field. A value of -1 indicates this criterion is not used for classification.	-1	2.5
PCPExclude	boolean	W	If *false*, the classification includes only those frames that match the *PCP* entry, if specified. If *true*, the classification includes all frames except those that match the *PCP* entry, if specified.	false	2.5
Device.X_84D32A_IEEE1905.AL.NetworkTopology.	object	-	This object represents the 1905.1 Network Topology capabilities of this device.	-	2.5
Enable	boolean	W	Enables or disables the 1905.1 Network Topology capabilities of device. When set to*true*, the device clears and (re)populates the *IEEE1905Device* and *ChangeLog* tables. When set to *false*, the contents of the *IEEE1905Device* and *ChangeLog* tables are implementation specific.	-	2.5
Status	string	-	When *Enable* is set to *true*, this parameter indicates the transient phase of the discovery of the *NetworkTopology*.	-	2.5

Name	Type	Write	Description	Object Default	Version
			• *Incomplete* (Indicates that the device is populating the topology object during the transient phase) • *Available* (Indicates that the transient phase is over and the device is maintaining and updating the topology object as changes occur) • *Error_Misconfigured* (Indicates that a necessary configuration value is undefined or invalid)		
MaxChangeLogEntries	unsignedInt-[1:]	W	The maximum number of entries allowed in the *ChangeLog* table.	-	2.5
LastChange	string(256)	-	The value MUST be the path name of a row in the *ChangeLog* table. If the referenced object is deleted, the parameter value MUST be set to an empty string. If the *ChangeLog*is modified the parameter is modified to reflect the last entry added to the *ChangeLog*.	-	2.5
IEEE1905DeviceNumberOfEntries	unsignedInt	-	The number of entries in the *IEEE1905Device* table.	-	2.5
ChangeLogNumberOfEntries	unsignedInt	-	The number of entries in the *ChangeLog* table.	-	2.5
Device.X_84D32A_IEEE1905.AL.NetworkTopology.ChangeLog.{i}.	object	-	This object represents log entries for changes in the 1905.1 Network Topology. The Change Log is a First In First Out queue where the oldest entries (defined by values of the *TimeStamp* parameter) are deleted once the log is full.	-	2.5
TimeStamp	dateTime	-	Date and Time at which the entry was added to the *ChangeLog* table.	-	2.5
EventType	string	-	Type of event for this entry. Enumeration of: • *NewNeighbor* (Entry represents a discovery of a Neighbor) • *LostNeighbor* (Entry represents the loss of a Neighbor)	-	2.5
ReporterDeviceId	string(17)	-	[*MACAddress*] 1905.1 AL MAC Address of device which reported the change.	-	2.5
ReporterInterfaceId	string(17)	-	[*MACAddress*] MAC Address of the interface of the reporting device on which the change has been detected.	-	2.5
NeighborType	string	-	Type of Neighbor for this event. Enumeration of: • *IEEE1905* • *Non-IEEE1905*	-	2.5
NeighborId	string(17)	-	[*MACAddress*] MAC Address of the Neighbor of this event. If the value of the *EventType*parameter is *NewNeighbor*, then the value represents the MAC Address of new Neighbor that joined the network; if the value of the *EventType* parameter is*LostNeighbor*, then the value represents the MAC Address of Neighbor that left the network.	-	2.5

Name	Type	Write	Description	Object Default	Version
			If value of the *NeighborType* parameter is *IEEE1905*, then this value of this parameter is the 1905.1 AL MAC Address of the Neighbor.		
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.	object	-	This object represents an instance of discovered 1905.1 Devices in the network. At most one entry in this table can exist with a given value for *IEEE1905Id*.	-	2.5
IEEE1905Id	string(17)	-	[*MACAddress*] 1905.1 AL MAC Address.	-	2.5
Version	unsignedInt	-	1905.1 Version Number.	-	2.5
InterfaceNumberOfEntries	unsignedInt	-	The number of entries in the *Interface* table.	-	2.5
NonIEEE1905Neighbor NumberOfEntries	unsignedInt	-	The number of entries in the *NonIEEE1905Neighboron* table.	-	2.5
IEEE1905NeighborNumberOf Entries	unsignedInt	-	The number of entries in the *IEEE1905Neighbor* table.	-	2.5
BridgingTupleNumberOf Entries	unsignedInt	-	The number of entries in the *BridgingTuple* table.	-	2.5
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.Interface.{i}.	object	-	This object represents an instance of an interface for the *IEEE1905Device*. At most one entry in this table can exist with a given value for *InterfaceId*.	-	2.5
InterfaceId	string(17)	-	[*MACAddress*] MAC Address of the interface.	-	2.5
MediaType	string	-	Media type of this *Interface*. Enumeration of: • *IEEE 802.3u* (IEEE 802.3u Fast Ethernet) • *IEEE 802.3ab* (IEEE 802.3ab Gigabit Ethernet) • *IEEE 802.11b* (IEEE 802.11b (2.4GHz)) • *IEEE 802.11g* (IEEE 802.11g (2.4GHz)) • *IEEE 802.11a* (IEEE 802.11a (5GHz)) • *IEEE 802.11n 2.4* (IEEE 802.11n (2.4GHz)) • *IEEE 802.11n 5.0* (IEEE 802.11n (5GHz)) • *IEEE 802.11ac* (IEEE 802.11ac (5GHz)) • *IEEE 802.11ad* (IEEE 802.11ad (60GHz)) • IEEE 802.11af (IEEE 802.11af) • *IEEE 1901 Wavelet* (IEEE 1901 Wavelet) • *IEEE 1901 FFT* (IEEE 1901 FFT) • *MoCAv1.1* (MoCAv1.1)	-	2.5
PowerState	string	-	The Power State of this *Interface*. Enumeration of: • *On* • *Power_Save*	-	2.5

Name	Type	Write	Description	Object Default	Version
			• *Off* • *Unsupported*		
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.NonIEEE1905Neighbor.{i}.	object	-	This object represents an instance of a Non-IEEE 1905 Neighbor for the *IEEE1905Device*. At most one entry in this table can exist with the same values for *LocalInterface* and *NeighborInterfaceId*.	-	2.5
LocalInterface	string(256)	-	The value MUST be the path name of a row in the *IEEE1905Device.{i}.Interface* table. If the referenced object is deleted, the parameter value MUST be set to an empty string.	-	2.5
NeighborInterfaceId	string(17)	-	[*MACAddress*] MAC Address of the interface for the *NonIEEE1905Neighbor*.	-	2.5
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.IEEE1905Neighbor.{i}.	object	-	This object represents an instance of an *IEEE1905Neighbor* for the *IEEE1905Device*. At most one entry in this table can exist with the same values for *LocalInterface* and *NeighborDeviceId*.	-	2.5
LocalInterface	string(256)	-	The value MUST be the path name of a row in the *IEEE1905Device.{i}.Interface* table. If the referenced object is deleted, the parameter value MUST be set to an empty string.	-	2.5
NeighborDeviceId	string(17)	-	[*MACAddress*] 1905.1 AL MAC Address of the Neighbor.	-	2.5
IEEE802dot1Bridge	boolean	-	If *true* then IEEE 802.1 Bridge(s) were detected between the local interface and the 1905.1 Neighbor.	-	2.5
Device.X_84D32A_IEEE1905.AL.NetworkTopology.IEEE1905Device.{i}.BridgingTuple.{i}.	object	-	This object represents an instance of an *BridgingTuple* for the *IEEE1905Device*.	-	2.5
InterfaceList	string(2048)	-	Comma-separated list (maximum length 2048) of strings (maximum length 256). Each list item MUST be the path name of a row in the *IEEE1905Device.{i}.Interface* table. If the referenced object is deleted, the corresponding item MUST be removed from the list.	-	2.5
Device.X_84D32A_IEEE1905.AL.Security.	object	-	This object represents the Security configuration for the 1905.1 Device as defined in [sub-clause 9.2 Security Setup Methods/IEEE1905.1].	-	2.5
SetupMethod	string	W	Security setup method for the network. Enumeration of: • *UCPK* (User Configured Passphrase or Key) • *PBC* (Push Button Configuration)	-	2.5
Password	string	W	1905.1 common password for generating security keys. When read, this parameter returns an empty string, regardless of the actual value.	-	2.5

C.4.1 Inform and notification requirements

For a 1905.1 device, all of the parameters listed in Table C-3 that are present in the data model implementation are required on every inform.

Table C-3—Forced inform parameters for a 1905.1 device

Parameter
Device.X_84D32A_IEEE1905.Version
Device.X_84D32A_IEEE1905.AL.IEEE1905Id

Active notification shall be enabled for all of the parameters listed in Table C-4 that are present in the data model implementation, regardless of the value of the notification attribute for these parameters. As a result, any change in the value of these parameters due to an entity other than the ACS *must* result in the CPE initiating a connection to the ACS to issue the inform method call.

Table C-4—Forced active notification parameters for a 1905.1 device

Parameter
Device.X_84D32A_IEEE1905.Version

A 1905.1 device shall support passive notification for all parameters defined in the 1905.1 device data model, with no exceptions.

C.5 Profiles

This subclause specifies the profiles defined for the 1905.1 device data model. The use of profiles for this data model follows the definition and usage conventions described in TR-106.

C.5.1 Notations

Table C-5 shows the abbreviations that are used to specify profile requirements.

Table C-5—Profile requirements

Abbreviation	Description
R	Read support is *required*.
W	Both read and write support is *required*. This *must not* be specified for a parameter that is defined as read only.
P	The object is *required* to be present.

C.5.1.1 X_84D32A_IEEE1905Baseline:1 profile

Table C-6 defines the X_84D32A_IEEE1905Baseline:1 profile for the X_84D32A_Device:2 data model. The minimum *required* version for this profile is X_84D32A_Device:2.5.

Table C-6—Baseline profile definition for 19051Device:1

Name	Requirement
Device.X_84D32A_IEEE1905.	P
Version	R
Device.X_84D32A_IEEE1905.AL.	P
IEEE1905Id	R

Name	Requirement
Status	R
LastChange	R
LowerLayers	R
InterfaceNumberOfEntries	R
Device.X_84D32A_IEEE1905.AL.Interface.{i}.	P
InterfaceId	R
Status	R
LastChange	R
LowerLayers	R
MediaType	R
Device.X_84D32A_IEEE1905.AL.Security.	P
SetupMethod	W
Password	W

C.5.1.2 X_84D32A_IEEE1905Power:1 profile

Table C-7 defines the X_84D32A_IEEE1905Power:1 profile for the X_84D32A_Device:2 data model. It is defined as the union of the X_84D32A_IEEE1905Baseline:1 profile and the additional requirements defined in this table. The minimum *required* version for this profile is X_84D32A_Device:2.5.

Table C-7—X_84D32A_IEEE1905Power:1

Name	Requirement
Device.X_84D32A_IEEE1905.AL.Interface.{i}.	P
PowerState	W
Device.X_84D32A_IEEE1905.AL.Interface.{i}.-VendorProperties.	P
OUI	R
Information	R

C.5.1.3 X_84D32A_IEEE1905InterfaceSelection:1 profile

Table C-8 defines the X_84D32A_IEEE1905InterfaceSelection:1 profile for the X_84D32A_Device:2 data model. It is defined as the union of the X_84D32A_IEEE1905Baseline:1 profile and the additional requirements defined in this table. The minimum *required* version for this profile is X_84D32A_Device:2.5.

Table C-8—X_84D32A_IEEE1905InterfaceSelection:1 profile

Name	Requirement
Device.X_84D32A_IEEE1905.AL.Interface.{i}.	P
PowerState	W
Device.X_84D32A_IEEE1905.AL.Interface.{i}.-VendorProperties.	P
OUI	R
Information	R
Device.X_84D32A_IEEE1905.AL.ForwardingTable.	P
ForwardingRuleNumberOfEntries	R
Device.X_84D32A_IEEE1905.AL.ForwardingTable.-	C

Name	Requirement
ForwardingRule.{i}.	
InterfaceList	W
MACDestinationAddress	W
MACDestinationAddressExclude	W
MACSourceAddress	W
MACSourceAddressExclude	W
EtherType	W
EtherTypeExclude	W
Vid	W
VidExclude	W
PCP	W
PCPExclude	W

C.5.1.4 X_84D32A_IEEE1905LinkMetric:1 profile

Table C-9 defines the X_84D32A_IEEE1905LinkMetric:1 profile for the X_84D32A_Device:2 data model. It is defined as the union of the X_84D32A_IEEE1905Baseline:1 profile and the additional requirements defined in this table. The minimum *required* version for this profile is X_84D32A_Device:2.5.

Table C-9—X_84D32A_IEEE1905LinkMetric:1 profile

Name	Requirement
Device.X_84D32A_IEEE1905.AL.Interface.{i}.	P
PowerState	W
LinkNumberOfEntries	R
Device.X_84D32A_IEEE1905.AL.Interface.{i}.-VendorProperties.	P
OUI	R
Information	R
Device.X_84D32A_IEEE1905.AL.Interface.{i}.Link.{i}.	P
InterfaceId	R
IEEE1905Id	R
MediaType	R
Device.X_84D32A_IEEE1905.AL.Interface.{i}.Link.{i}.-Metric.	P
PacketErrors	R
TransmittedPackets	R
PacketsReceived	R
MACThroughputCapacity	R
LinkAvailability	R
PHYRate	R
RSSI	R

C.5.1.6 Network topology profile

Table C-10 defines the network topology profile for the 19051Device:1 object.

Table C-10—X_84D32A_IEEE1905NetworkTopology:1 profile

Name	Requirement
Device.X_84D32A_IEEE1905.AL.NetworkTopology.	P
Enable	W
Status	R
MaxChangeLogEntries	W
LastChange	R
IEEE1905DeviceNumberOfEntries	R
ChangeLogNumberOfEntries	R
Device.X_84D32A_IEEE1905.AL.NetworkTopology.-ChangeLog.{i}.	P
TimeStamp	R
EventType	R
ReporterDeviceId	R
ReporterInterfaceId	R
NeighborType	R
NeighborId	R
Device.X_84D32A_IEEE1905.AL.NetworkTopology.-IEEE1905Device.{i}.	P
IEEE1905Id	R
Version	R
InterfaceNumberOfEntries	R
NonIEEE1905NeighborNumberOfEntries	R
IEEE1905NeighborNumberOfEntries	R
BridgingTupleNumberOfEntries	R
Device.X_84D32A_IEEE1905.AL.NetworkTopology.-IEEE1905Device.{i}.BridgingTuple.{i}.	P
InterfaceList	R
Device.X_84D32A_IEEE1905.AL.NetworkTopology.-IEEE1905Device.{i}.IEEE1905Neighbor.{i}.	P
LocalInterface	R
NeighborDeviceId	R
IEEE802dot1Bridge	R
Device.X_84D32A_IEEE1905.AL.NetworkTopology.-IEEE1905Device.{i}.Interface.{i}.	P
InterfaceId	R
MediaType	R
PowerState	R
Device.X_84D32A_IEEE1905.AL.NetworkTopology.-IEEE1905Device.{i}.NonIEEE1905Neighbor.{i}.	P
LocalInterface	R
NeighborInterfaceId	R