IEEE P802.15
Wireless Personal Area Networks

Project	IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)
Title	MedWiN MAC and Security Proposal
Date Submitted	May 4, 2009
Source	David Davenport (1), Neal Seidl (2), Jeremy Moss (3), Maulin Patel (4), Anuj Batra (5), Jin-Meng Ho (5), Srinath Hosur (5), JuneChul Roh (5), Tim Schmidl (5), Okundu Omeni (6), Alan Wong (6) (1) GE Global Research, davenport@research.ge.com, 518-387-5041, 1 Research Circle, Niskayuna, NY, USA (2) GE Healthcare, neal.seidl@med.ge.com, 414-362-3413, 8200 West Tower Avenue, Milwaukee, WI, USA (3) Philips, j.moss@philips.com, +44 1223 427530, 101 Cambridge Science Park, Milton Road, Cambridge UK (4) Philips, maulin.patel@philips.com, 914-945-6156, 345 Scarborough Road, Briarcliff Manor, NY, USA (5) Texas Instruments, {batra@ti.com, 214-480-4220}, {jinmengho@ti.com, 214-480-1994}, {hosur@ti.com, 214-480-4432}, {jroh@ti.com, 214-567-4145}, {schmidl@ti.com, 214-480-4460}, 12500 TI Blvd, Dallas, TX, USA (6) Toumaz Technology, {okundu.omeni@tomuaz.com, +44 1235 438950}, {alan.wong@toumaz.com, +44 1235 438961}, Building 3, 115 Milton Park, Abingdon, Oxfordshire, UK
Re:	Response to IEEE 802.15.6 call for proposals
Abstract	This submission provides the normative text for a joint MAC and security proposal presented in two accompanying documents doc. IEEE 802.15-09-0326-00-0006 and doc. IEEE 802.15-09-0325-00-0006.
Purpose	To submit a joint proposal on MAC and security to the IEEE 802.15.6 task group
Notice	
Release	

TABLE OF CONTENTS

1 Scope

This standard specifies a medium access control (MAC) sublayer in support of a physical layer for wireless body area networks (BANs). It also specifies message security services provided at the MAC sublayer and security key generations performed inside or/and outside the MAC sublayer.

2 References

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

IEEE Std 802-2001, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.[1]

IEEE Std P1363-2000, IEEE Standard Specifications for Public-Key Cryptography.

FIPS Pub 197, Advanced Encryption Standard (AES), November 2001. [2]

FIPS Pub 186-3 (Draft), Digital Signature Standard (DSS), November 2008.

NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005.

NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, July 2007.

NIST Special Publication 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, March 2007.

ISO/IEC 10646, Universal Multiple-Octet Coded Character Set (UCS), December 2003. Amendment 1, November 2005. Amendment 2, July 2006. Amendment 3, February 2008.[3]

IETF RFC 4086, Randomness Requirements for Security, June 2005.[4]

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

[2] FIPS publications and NIST special publications are available from the National Institute for Standards and Technology, 100 Bureau Drive, Stop 8900, Gaithersburg, MD 20899-8900 (http://www.nist.gov).

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

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

3 Definitions

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

allocation: One or more time intervals that a node or a hub obtains using an access method to initiate one or more frame transactions. An allocation comprises one or more allocation intervals. Reference to allocation of a node means that the node is the sender or recipient in the allocation.

allocation direction: Reference to an uplink allocation or a downlink allocation, but not both, obtained by a node or a hub, respectively, to initiate one or more frame transactions.

allocation interval: A continuous time interval in an allocation, comprising one or more consecutive allocation slots. Reference to allocation interval of a node means that the node is the sender or recipient in the allocation interval.

allocation slot: A time unit used to designate the lengths of medium access related time intervals, such as beacon period, allocation interval, and random access phase (RAP).

association: A procedure used to identify a node and a hub to each other and establish a new master key shared between them or activate an existing master key pre-shared between them.

authentication: A measure used to ensure that the identity of the node or the hub in an association procedure is not one forged by a third party.

beacon: A frame transmitted by a hub to facilitate subnet management, such as the coordination of medium access and power management of the nodes in the subnet of the hub, and to facilitate clock synchronization within the subnet.

beacon period: A repetitive time interval of the same nominal duration in which a hub transmits a beacon.

contended allocation: A non-reoccurring time interval, within a random access phase (RAP), that a node obtains using random access to initiate a frame transaction. A contended allocation is an uplink allocation, suitable for "unpredictable" uplink traffic (for example, due to data rate variations and/or channel impairments).

connected node: A node that has a connection with a hub.

connection: A relationship between a node and a hub in the same subnet, substantiated by a Connected_NID assigned to the node by the hub, and by a power management profile and an access allocation contract negotiated between the node and the hub.

downlink: A communications link for transfer of management and data traffic from a hub to a node.

downlink allocation: An allocation in which a hub initiates a frame transaction to transmit management and data traffic to a node and the node returns acknowledgment if required.

frame: An uninterrupted sequence of octets delivered by the medium access control (MAC) sublayer to the physical (PHY) layer, by the PHY layer to the local transmit antenna, or vice versa.

frame transaction: Exchange of one or more functionally related frames between a node and a hub, without interruption by other frames to facilitate frame transmission and reception as well as frame acknowledgment if required.

hub: An entity that possesses a node's functionality and coordinates the medium access and power management of the nodes in its subnet.

hub identifier (HID): An abbreviated address of a hub.

improvised access: An access method, based on impromptu polling or posting by a hub, by which a node or a hub obtains a polled or posted allocation, typically located outside scheduled allocations, to initiate one or more frame transactions.

master key (MK): A secret bit string established or activated between a node and a hub in an association procedure and used to create a pairwise temporal key (PTK) shared between them.

multi-periodic (m-periodic) allocation: A scheduled allocation that has allocation intervals reoccurring in every mth beacon period with m being an integer larger than one. An m-periodic allocation is either an uplink allocation or a downlink allocation, suitable for low duty cycle periodic or quasi-periodic traffic.

node: An entity that contains a medium access control (MAC) sublayer and a physical (PHY) layer and provides message security services in accordance with this standard. Unless otherwise noted, reference to nodes refers to nodes that are not acting as hubs.

node identifier (NID): An abbreviated address of a single node or of a logical group of nodes.

non-secure frame: A term that is interchangeable with unsecured frame.

one-periodic (1-periodic) allocation: A scheduled allocation that has allocation intervals reoccurring in every beacon period. A 1-periodic allocation is either an uplink allocation or a downlink allocation, suitable for high duty cycle periodic or quasi-periodic traffic.

pairwise temporal key (PTK): A secret bit string shared between a node and a hub and used to secure frames transferred between them.

pairwise temporal key (PTK) creation: A procedure used to create a PTK between a node and a hub based on a master key shared between them, and to confirm possession of a shared MK between the node and the hub.

poll: A control frame sent by the hub to grant an immediate polled allocation to the addressed node or to inform the node of a future poll.

polled allocation: A non-reoccurring time interval, typically outside scheduled allocations, that a node obtains using improvised access to initiate a frame transaction. A polled allocation is an uplink allocation, suitable for "unexpected" or "extra" uplink traffic (for example, due to data rate variations and/or channel impairments).

post: A contention-free frame transaction initiated by a hub with a node within its subnet outside a scheduled allocation. A post starts a posted allocation.

posted allocation: A non-reoccurring time interval, typically outside scheduled allocations, that a hub obtains using improvised access to initiate a contention-free frame transaction with a node within its subnet. A posted allocation is a downlink allocation, suitable for "unexpected" or "extra" downlink traffic (for example, due to network management needs, data rate variations, and/or channel impairments).

random access: An access method, based on carrier sense multiple access with collision avoidance (CSMA/CA), by which a node obtains a time interval in a random access phase (RAP) to initiate one or more frame transactions.

random access phase (RAP): A time span set aside by a hub for random access to the medium by the nodes in the subnet of the hub.

scheduled access: An access method, based on advance reservation and scheduling, by which a node or a hub obtains one or more reoccurring time intervals to initiate frame transactions.

scheduled allocation: One or more reoccurring time intervals that a node or a hub obtains using scheduled access to initiate frame transactions. A scheduled allocation is either an uplink allocation or a downlink allocation, suitable for high or low duty cycle periodic or quasi-periodic traffic.

secured communication: Exchange of secured frames.

secure frame: A term that is interchangeable with secured frame.

secured frame: A frame that is secured with authenticity, integrity, confidentiality if required, and replay protection.

subnet: A logical network partition comprising a hub and one or more nodes whose medium access and power management are coordinated by the hub.

uplink: A communications link for transfer of management and data traffic from a node to a hub.

uplink allocation: An allocation in which a node initiates a frame transaction to transmit management and data traffic to a hub and the hub returns acknowledgment if required.

unsecured communication: Exchange of unsecured frames.

unsecured frame: A frame that is not secured with authenticity, integrity, confidentiality, or replay protection.

4 Acronyms and abbreviations

AES	Advanced Encryption Standard
APN	association protocol number
CBC	cipher block chaining
CCA	clear channel assessment
CCM	counter mode for message encryption and cipher block chaining for message integrity protection
CMAC	(block) cipher-based message authentication code algorithm
CRC	cyclic redundancy check
CSMA/CA	carrier sense multiple access with collision avoidance
FCS	frame check sequence
GT	guardtime
GTK	group temporal key
HID	hub identifier
HME	hub management entity
IE	information element
KCK	key confirmation key
KMAC	keyed message authentication check
LFSR	linear feedback shift register
MAC	medium access control
MIC	message integrity code
MK	master key
MSDU	MAC service data unit
NID	node identifier
NME	node management entity
PHY	physical layer
PN	pseudo-random
PTK	pairwise temporal key

RAP	random access phase
TIFS	turnaround inter-frame space
TK	temporal key
UP	user priority

5 General framework elements

This clause provides the basic framework required of all nodes and hubs. The framework serves as a prerequisite to supporting the functions of nodes and hubs and their interactions specified later in detail in this standard. It covers four fundamental aspects— the network topology used for medium access, the reference model used for functional partitioning, the state diagram used for frame exchange, and the security paradigm used for message protection.

5.1 Network topology

All nodes and hubs shall be organized into logical sets, referred to as subnets in this specification, and coordinated by their respective hubs for medium access and power management as illustrated in Figure 1. There shall be one and only one hub in a subnet, whereas the number of nodes in a subnet may range from zero to mMaxNumberSubnet. Frame exchanges may occur directly only between nodes and the hub within the same subnet.

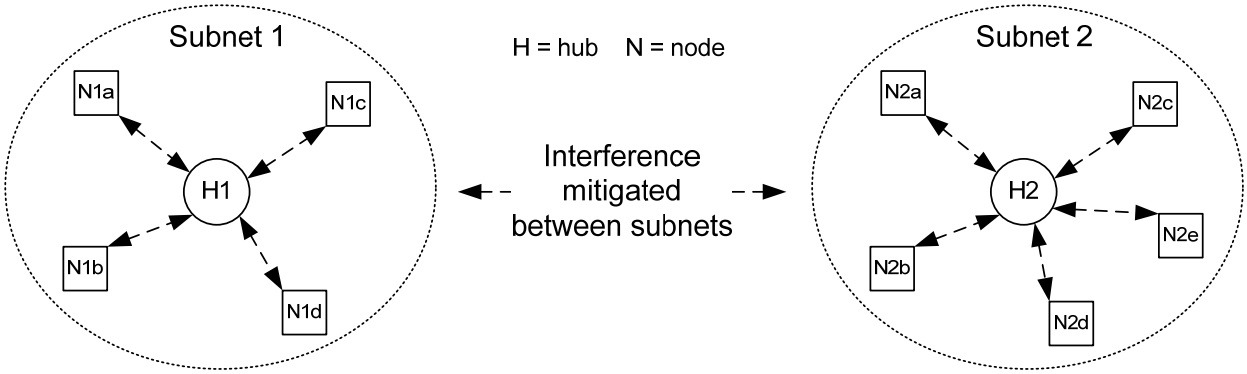


Figure 1 — Network topology

Access coordination at the MAC sublayer between subnets is not specified in this standard. Nodes referenced in this standard are in the context of a given subnet, unless noted otherwise. Two possible mechanisms for mitigating the interference between adjacent or overlapping subnets are provided through pseudo-random shifting of reference times across beacon periods and pseudo-random hopping of channels over time scales of multiple beacon periods.

5.2 Reference model

All nodes and hubs shall be internally partitioned into a physical (PHY) layer and a medium access control (MAC) sublayer, in accordance with the ISO/OSI-IEEE 802 reference model, as shown in Figure 2. Direct communications between a node and a hub shall transpire at the PHY layer and MAC sublayer as specified in this standard; the PHY and MAC sublayer of a node or a hub shall operate in one channel at any given time. Message security services shall occur at the MAC sublayer, and security key generations may take place inside and/or outside the MAC sublayer.

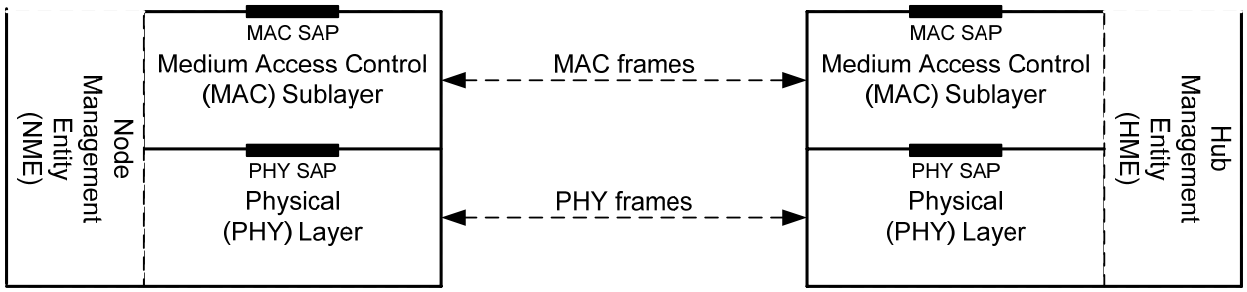
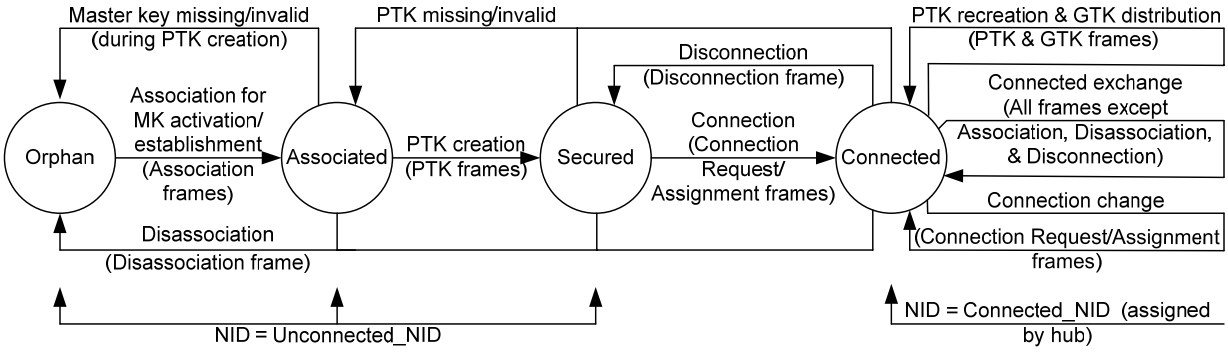


Figure 2 — Reference model

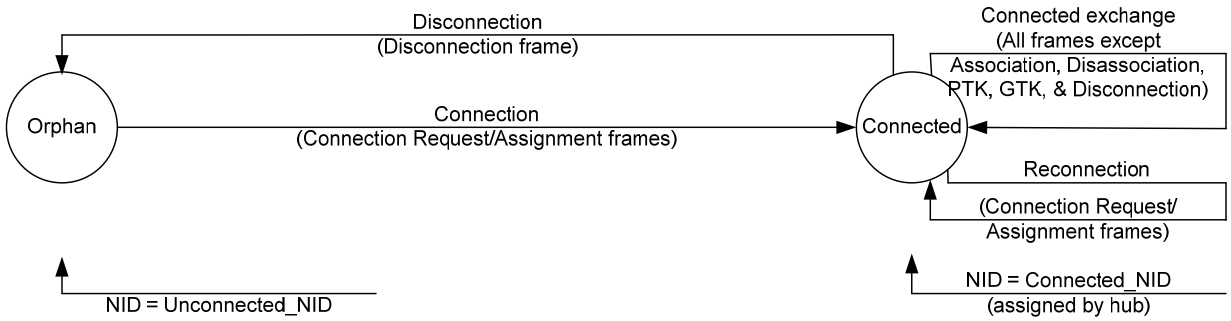
Within a node or a hub, the PHY provides its service to the MAC through the PHY service access point (SAP) located between them, and the MAC provides its service to the MAC client through the MAC SAP located immediately above the MAC sublayer. There may be a logical node management entity (NME) or hub management entity (HME) that exchanges network management information with the PHY and MAC as well as with other layers. The HME is a superset of the NME in terms of the management functionality they each support. However, the presence of the NME or HME and the partitioning between the NME or HME and the MAC or the PHY is not mandated, nor is the behavior of the NME or HME specified, in this standard.

5.3 State diagram

All nodes and hubs shall go through certain stages, i.e., states, at the MAC level before they exchange user (MAC client) data, as shown in Figure 3, where frames permitted or required to exchange at each state are also indicated. State classification and transition is defined with respect to a pair of a node and a hub, but is often referenced in the name of the node only. Beacon frames, which are not shown in Figure 3 or mentioned in the rest of 5.3, shall be transmitted by hubs in secured (authenticated but not encrypted) or unsecured (neither authenticated nor encrypted) frames, regardless of which states a hub is in conjunction with one node or another.



(a) Secured communication



(b) Unsecured communication

Figure 3 — Access state diagram

5.3.1 Secured communication

A node and a hub shall follow the MAC state diagram Figure 3(a) for secured communication if either of them requires secured frame exchanges with the other.

5.3.1.1 Orphan state

At this state, the node does not have any relationship with a hub for secured communication. It is the state that the node initially enters in relation with a hub. The node and a hub shall not transmit any frames to each other —other than Association and control unsecured frames. The node may find a hub from received beacons, and may exchange Association frames with the hub to establish an association, i.e., to activate a pre-shared MK or generate a new master key (MK), and to authenticate with each other if so required, thereby transitioning to the next state, the Associated state. However, if the node and the hub fail to activate or establish a shared MK, they shall not advance to the Associated state.

5.3.1.2 Associated state

At this state, the node is associated, i.e., holds a shared master key, with a hub for PTK creation. The node and the hub shall not transmit any frames to each other —other than Disassociation, Pairwise Temporal Key (PTK), and control unsecured frames. The node may exchange PTK frames with the hub to establish a secured relationship, i.e., to verify possession of a shared MK and to create a PTK, thereby transitioning to the next state, the Secured state. However, if the node and the hub fail to create a PTK, they shall not advance to the Secured state. To repeal an association and hence the current MK, either the node or the hub may send a Disassociation frame to the other, thus moving back to the Orphan state.

5.3.1.3 Secured state

At this state, the node is secured, i.e., holds a PTK, with a hub for message security, i.e., secured frame exchanges. The node and the hub shall not transmit any frames to each other—other than Disassociation, Connection Request, and Connection Assignment secured frames and control unsecured frames. The node may exchange Connection Request and Connection Assignment frames with the hub to establish a connection, thereby transitioning to the next and final state, the Connected state. However, if the node and the hub fail to establish a connection, they shall not advance to the Connected state.

5.3.1.4 Connected state

At this state, the node is connected, i.e., holds an assigned Connected_NID, a negotiated power management profile, and an access allocation contract, with a hub for abbreviated node addressing as well as desired wakeup and allocation arrangement. The node and the hub may transmit any secured frames to each other—other than

Association secured frames, but shall not transmit any unsecured frames to each other—other than control unsecured frames. To repeal the power management profile and access allocation contract as well as the Connected_NID of the node, either the node or the hub may send a Disconnection frame to the other, thereby moving back to the Secured state.

5.3.2 Unsecured communication

A node and a hub shall follow the MAC state diagram Figure 3(b) for unsecured communication if neither of them requires secured frame exchanges with the other.

5.3.2.1 Orphan state

At this state, the node does not have any relationship with a hub for unsecured communication. It is the state that the node initially enters in relation with a hub. The node and the hub shall not transmit any frames to each other—other than Connection Request, Connection Assignment, and control unsecured frames. The node may exchange Connection Request and Connection Assignment frames with the hub to establish a connection, thereby transitioning to the next and final state, the Connected state. However, if the node and the hub fail to establish a connection, they shall not advance to the Connected state.

5.3.2.2 Connected state

At this state, the node is connected, i.e., holds an assigned Connected_NID, a negotiated power management profile, and an access allocation contract, with a hub for abbreviated node addressing as well desired wakeup and allocation arrangement. The node and the hub may transmit any unsecured frames to each other—other than Association, Disassociation, and PTK frames, but shall not transmit secured frames to each other. To change to secured communications between them, the node and the hub shall disconnect from each other, thus moving back to the Orphan state and then following the state diagram Figure 3(a) for secured communication. To repeal the power management profile and access allocation contract as well as the Connected_NID of the node, either the node or the hub may send a Disconnection frame to the other, thereby moving back to the Orphan state.

5.4 Security paradigm

All nodes and hubs shall be offered three security levels in this standard:

– Level 0 – unsecured communication. At this level, messages are transmitted in unsecured frames, which provide no measures for message authenticity and integrity validation, confidentiality and privacy protection, and replay defense.

– Level 1 – authentication but not encryption. At this level, messages are transmitted in secured authenticated but not encrypted frames, which provide measures for message authenticity and integrity validation and replay defense but not confidentiality and privacy protection.

– Level 2 – authentication and encryption. At this level, messages are transmitted in secured authenticated and encrypted frames, which provide measures for message authenticity and integrity validation, confidentiality and privacy protection, and replay defense.

A node and a hub shall jointly select a security level suitable for their frame exchanges, based on their respective security requirements and certain information specific to each other. The security selection shall be facilitated by the security requirement and security support information indicated by the node and the hub, respectively.

The node and the hub shall create a pairwise temporal key (PTK) for their unicast secured communication. The hub shall distribute a group temporal key (GTK) for its broadcast or multicast secured communication.

The node and the hub shall activate a pre-shared master key (MK) or establish a new MK by unauthenticated or authenticated association for the PTK creation.

The node and the hub shall follow the security relationships shown in Figure 4 to provide message security services and perform security key generations for secured communication. A “session” indicated in this figure refers to a time span in which a temporal key remains valid. The length of the “session” is determined by the security policy

governing data transfers between the two communicating parties. It is further limited by the technical restrictions on the reuse of the same temporal key over successive messages (i.e., MAC frames in this specification).

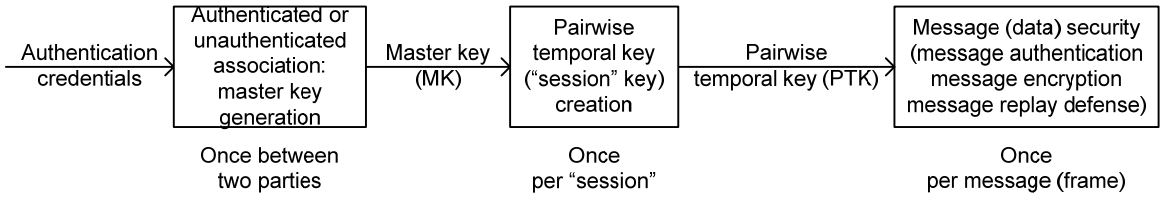


Figure 4 — Security hierarchy

6 MAC frame formats

This clause defines the formats of MAC frames. It starts in 6.1 with the general conventions for specifying the transmit order of MAC frames and their constituent fields, followed in 6.2 by the format definition for the common fields that are present in all MAC frames. The next three subclauses 6.3-6.5 define each frame type and subtype in detail. The final subclause 6.6 defines a number of information elements that appear in certain MAC frames.

6.1 Conventions

A MAC frame is an ordered sequence of fields delivered to or from the PHY SAP. Each figure in Clause 6 depicts the fields contained in a MAC frame, or a part thereof, from left to right in the transmit order. The figure also indicates the number of octets contained in each field and the corresponding octet transmit order, on top of the field, as illustrated in Figure 5.

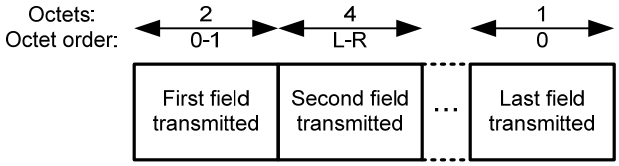


Figure 5 — Example figure depicting fields aligned with octet boundaries

Unless otherwise noted, an atomic field, i.e., a field that is not in turn comprised of other fields, denotes a numerical value encoded in unsigned binary. If such a field contains F octets (F > 1), octet 0 is the octet containing the least-significant bits of that field and is the first octet transmitted of the field, whereas octet F-1 is the octet containing the most-significant bits of that field and is the last octet transmitted of the field. The octet order is indicated as "L-R", i.e., from left to right, above a multi-octet non-atomic field.

In a figure that depicts certain fields not aligned with octet boundaries, the number of bits and the corresponding bit order of encoding are shown instead for each field in the figure, as illustrated in Figure 6. Bits are ordered continually across the fields that are not aligned with octet boundaries, from left to right, starting from bit 0, i.e., the least-significant bit of the bits comprising these fields. Bit numbering restarts from 0 in fields located on octet boundaries.

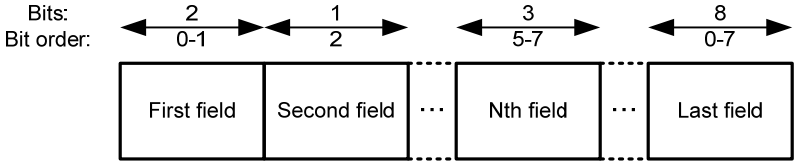


Figure 6 — Example figure depicting fields not aligned with octet boundaries

Each field is defined, or set, based on the perspective of the node or the hub that is sending the frame containing that field, referred to as the sender. It is parsed based on its definition by the hub or the node intended to receive the frame containing it, referred to as the recipient.

Reserved fields are set to zero on transmission and ignored on reception. If some values in a field are reserved, the field is not set to any of those reserved values on transmission. Unless otherwise noted, fields that are set to reserved values or defined based on other fields that are set to reserved values are ignored on reception.

6.2 General format

A MAC frame consists of a fixed-length MAC header, a variable-length MAC frame body, and a fixed-length FCS field as shown in Figure 7. The MAC frame body has a length *L_FB* such that $0 \leq L\_FB \leq$ mMaxFrameBodyLength, and is present only if it has a nonzero length.

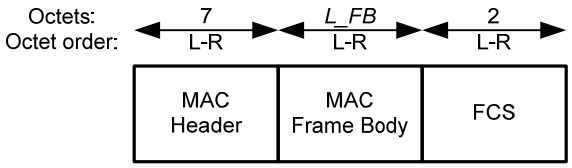


Figure 7 — MAC frame format

6.2.1 MAC header

The MAC header is formatted as shown in Figure 8.

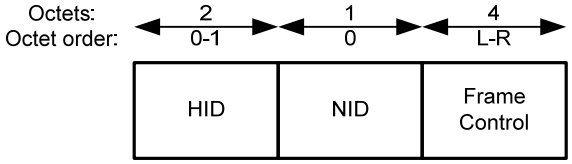


Figure 8 — MAC header format

6.2.1.1 HID

The HID field is set to the two octets containing the 16 least-significant bits of the IEEE MAC address included in the beacon.of the hub that is the sender or recipient of the current frame.

6.2.1.2 NID

The NID field is set according to Table 11 such that it identifies the node that is the sender or recipient of the current frame.

6.2.1.3 Frame Control

The Frame Control is formatted as shown in Figure 9.

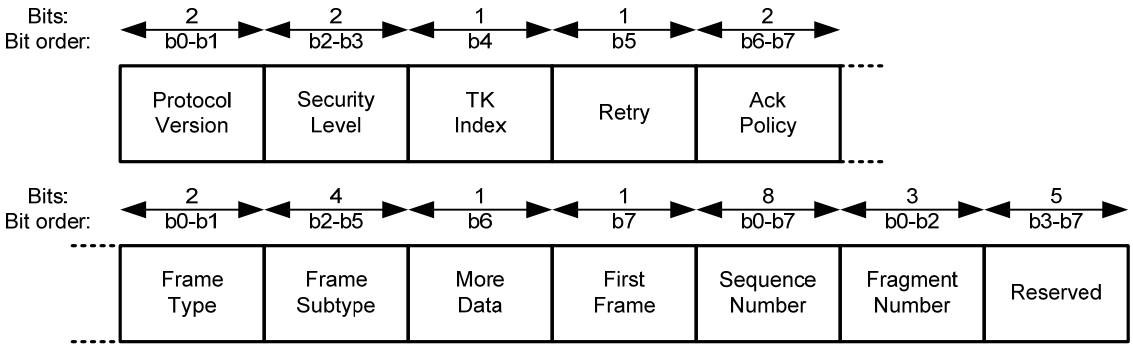


Figure 9 — Frame Control format

6.2.1.3.1 Protocol Version

The Protocol Version field is set to 0 for this revision of the standard. All other values are reserved. This field is invariant in size and place across all revisions of this standard.

6.2.1.3.2 Security Level

The Security Level field is set according to Table 1 such that it indicates the security level of the current frame.

Table 1 — Security Level field encoding

Field value b2 b3	Security level of current frame
00	Level 0 – frame not secured
10	Level 1 – frame authenticated but not encrypted
01	Level 2 – frame authenticated and encrypted
11	Reserved

6.2.1.3.3 TK Index

The TK Index field is set as follows to indicate the pairwise temporal key (PTK) or group temporal key (GTK) being used to secure the current frame:

– In frames secured by a PTK, it is set to the value of the PTK Index field in the PTK frames that were exchanged in creating the PTK.

– In frames secured by a GTK, it is set to the value of the GTK Index field in the GTK frame that was exchanged in distributing the GTK.

The TK Index field is reserved in unsecured frames.

6.2.1.3.4 Retry

The Retry field is set to 1 in any non-beacon management or data type frame that was transmitted previously and is being transmitted again. It is reserved in all other frames.

6.2.1.3.5 Acknowledgment (Ack) Policy

The Ack Policy field is set according to Table 2 to indicate the acknowledgement requirement of the current frame.

Table 2 — Acknowledgment (Ack) Policy field encoding

Field value b6 b7	Acknowledgment requirement
00	No acknowledgment (N-Ack)
10	Immediate acknowledgment (I-Ack)
11	Block acknowledgment later (L-Ack)
01	Block acknowledgment (B-Ack)

6.2.1.3.6 Frame Type

The Frame Type field is set to indicate the type of the current frame according to Table 3.

Table 3 — Frame Type and Frame Subtype field encoding

Frame Type value b0 b1	Frame Type name	Frame Subtype value b2 b3 b4 b5	Frame Subtype name
00	Management	0000	Beacon
00	Management	1000	Reserved
00	Management	0100	Association
00	Management	1100	Disassociation
00	Management	0010	PTK
00	Management	1010	GTK
00	Management	0110-1110	Reserved
00	Management	0001	Connection Request
00	Management	1001	Connection Assignment
00	Management	0101	Disconnection
00	Management	1101-1111	Reserved
10	Control	0000	I-Ack
10	Control	1000	B-Ack
10	Control	0100	I-Ack+Poll
10	Control	1100	B-Ack+Poll
10	Control	0010	Poll
10	Control	1010-1111	Reserved
01	Data	0000-1111	User defined data subtype
11	Reserved	0000-1111	Reserved

6.2.1.3.7 Frame Subtype

The Frame Subtype field is set to indicate the subtype of the current frame of a given type according to Table 3.

6.2.1.3.8 More Data

The More Data field is set as follows:

– In frames sent by a node to the hub,
  – it is set 0 if the node has no pending frame transactions to initiate with the hub, or
  – it is set to 1 if the node has one or more pending frame transactions to initiate with the hub.
– In non-beacon management and data type frames sent by a hub to a node,
  – it is set to 0 if the hub is not to send a post to the node after the current frame transaction, or

  - it is set to 1 if the hub is to send a post to the node TIFS after the end of the current frame transaction.
- In I-Ack and B-Ack frames sent by a hub to a node,
  - it is set to 0 if the hub is not to send a post to the node after the end of the current allocation interval, or
  - it is set to 1 if the hub is to send a post to the node at or after the end of the current allocation interval. In the latter case, when the hub is to send a post to the node is indicated in the Sequence Number and Fragment Number fields of the current frame.
- In Poll, I-Ack+Poll, and B-Ack+Poll frames sent by a hub to a node,
  - it is set to 0 if the hub grants to the node through this frame a polled allocation starting TIFS after the end of the current frame, with the end of the polled allocation indicated in the Sequence Number field of the current frame, or
  - it is set to 1 if the hub grants to the node through this frame no polled allocation but is to send another poll to the node at a future time as indicated in the Sequence Number and Fragment Number fields of the current frame.
- In all other frames sent by a hub to a node, it is reserved.

6.2.1.3.9 First Frame

The First Frame field is set to 1 if this is the first frame sent in a scheduled allocation interval. It is set to 0 otherwise.

6.2.1.3.10 Sequence Number

The Sequence Number field is set as follows:

- In beacon frames, it is incremented by one from its value in the last transmitted beacon.
- In non-beacon management type frames, it is reserved.
- In I-Ack and B-Ack frames sent by a hub to a node,
  - if the More Data field of the current frame is set to 0, it is reserved;
  - if the More Data field of the current frame is set to 1, it is set to *A* such that the hub is to send a post to the node at the start of the allocation slot numbered *A*, in a beacon period indicated in the Fragment Number field of the current frame.
- In I-Ack and B-Ack frames sent by a node to a hub, it is reserved.
- In Poll, I-Ack+Poll, and B-Ack+Poll frames sent by a hub to a node,
  - if the More Data field of the current frame is set to 0, indicating that the node is granted via this frame a polled allocation starting TIFS after the end of this frame, it is set to *A* such that the polled allocation ends at the end of the allocation slot numbered *A* in the current beacon period;
  - if the More Data field of the current frame is set to 1, indicating that the node is granted via this frame no polled allocation but is to be sent another poll, it is set to *A* such that the hub is to send the next poll at the start of the allocation slot numbered *A*, in a beacon period indicated in the Fragment Number field of the current frame.
- In data type frames,
  - it is incremented by one from its value in the frame that was of the same subtype and addressed to the same recipient(s) and that contained the previous MSDU or part thereof;
  - it has the same value in frames containing fragments of the same MSDU.

6.2.1.3.11 Fragment Number

The Fragment Number field is set as follows:

- In beacon frames,
  - it is set to 7 if the hub sending the current beacon frame is not to hop to another channel in the next 7 beacon periods not counting the current beacon period;
  - it is set to $H < 7$ if the hub sending the current beacon frame is to hop to the next channel in its channel hopping sequence at the end of the next Hth beacon period not counting the current beacon period if $H > 0$ or at the end of the current beacon period if $H = 0$.
- In non-beacon management type frames containing a frame payload not fragmented or the first fragment of a fragmented frame payload, it is set to 0.
- In non-beacon management type frames containing a non-first fragment of a fragmented frame payload, it is incremented by one from its value in the frame containing the previous fragment of the frame payload.
- In data type frames containing an MSDU not fragmented or the first fragment of a fragmented MSDU, it is set to 0.
- In data type frames containing a non-first fragment of a fragmented MSDU, it is incremented by one from its value in the frame containing the previous fragment of the MSDU.
- In I-Ack and B-Ack frames sent by a hub to a node,
  - if the More Data field of the current frame is set to 0, it is reserved;
  - if the More Data field of the current frame is set to 1, indicating that the hub is to send a post to the node at the start of the allocation slot indicated in the Sequence Number field of the current frame, it is set to B such that the allocation slot is the one located in the current beacon period if $B = 0$ or in the next Bth beacon period not counting the current beacon period if $B > 0$.
- In I-Ack and B-Ack frames sent by a node to a hub, it is reserved.
- In Poll, I-Ack+Poll, and B-Ack+Poll frames sent by a hub to a node,
  - if the More Data field of the current frame is set to 0, indicating that the node is granted via this frame a polled allocation starting TIFS after the end of this frame and ending at the end of the allocation slot indicated in the Sequence Number field of the current frame, it is reserved;
  - if the More Data field of the current frame is set to 1, indicating that the node is granted via this frame no polled allocation but is to be sent another poll at the start of the allocation slot indicated in the Sequence Number field of the current frame, it is set to B such that the allocation slot is the one located in the current beacon period if $B = 0$ or in the next Bth beacon period not counting the current beacon period if $B > 0$.

6.2.2 MAC frame body

The MAC frame body, when it has a nonzero length, is formatted as shown in Figure 10. The length of the MAC frame body *L_FB* must not exceed mMaxFrameBodyLength.

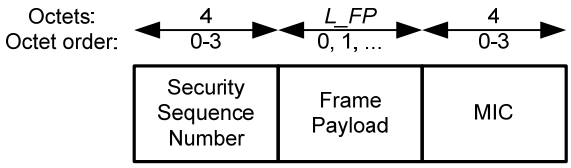


Figure 10 — MAC frame body format

The Security Sequence Number and Message Integrity Code (MIC) fields are not present in unsecured frames, as indicated by the Security Level field of the MAC header of the same frame.

6.2.2.1 Security Sequence Number

The Security Sequence Number field is set as follows for nonce construction and replay detection:

– In frames secured with a new pairwise temporal key (PTK) or group temporal key (GTK), it is set to 1.

– In frames secured with a used PTK, whether they are transmitted for the first time or they are retries, it is incremented by one from its value in the last transmitted frame secured with the same PTK.

– In frames secured with a used GTK, whether they are transmitted for the first time or they are retries, it is incremented by one from its value in the last transmitted frame secured with the same GTK.

The value of the Security Sequence Number field increments in frames secured with the same PTK or GTK, rather than in frames of the same frame type or frame subtype. It increments even if the current frame transmission is a retransmission of an earlier transmission.

6.2.2.2 Frame Payload

The Frame Payload field is set as follows:

– In management type frames not encrypted, it is set to a sequence of fields to be communicated to the recipient(s), with the fields defined in 6.3.

– In control type frames, it is set to a sequence of fields to be communicated to the recipient(s), with the fields defined in 6.4.

– In data type frames not encrypted, it is set to a sequence of octets passed as an MSDU through the MAC SAP to the MAC, without altering the order of the sequence.

– In management or data type frames encrypted, it is set to the encrypted frame payload, i.e., the cipher text of the frame payload that would otherwise be communicated in a frame not encrypted.

If a frame payload is fragmented and carried in multiple frames, the Frame Payload field is set to a fragment of the otherwise unfragmented frame payload.

The length of the Frame Payload field, denoted as *L_FP* in Figure 10, must be such that the length of the MAC frame body does not exceed mMaxFrameBodyLength. If the Frame Payload has a zero length, i.e., if a frame does not have a Frame Payload, then the frame does not have a MAC frame body either.

6.2.2.3 MIC

The MIC field is set to a keyed message authentication check (KMAC) for preserving the authenticity and integrity of the MAC header and the MAC frame body of the current secured frame, as further specified in 8.2.

6.2.3 FCS

The FCS field is formatted as shown in Figure 11, where a_{15} is the least-significant of the field, and a_0 is the most-significant bit. The bits a_{15}, a_{14}, …, a_0 are the binary coefficients of a CRC polynomial of degree 15 denoted as

$$R(x) = a_{15}x^{15} + a_{14}x^{14} + \ldots + a_1x + a_0$$

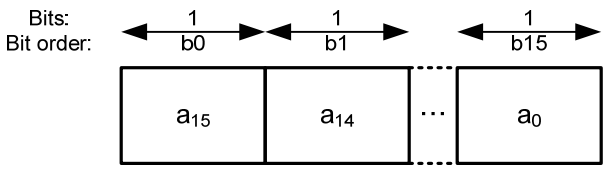


Figure 11 — FCS format

The CRC polynomial is calculated over a calculation field using the following CRC-16-CCITT standard generator polynomial of degree 16:

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

The calculation field is the MAC frame except the FCS field for this specification. It is mapped to a message polynomial M(x) of degree k-1, where k is the number of bits in the calculation field. The least-significant bit of the first octet presented to the PHY SAP is the coefficient of the x^{k-1} term, the next least-significant bit is the coefficient of the x^{k-2} term, …, and the most-significant bit of the last octet transmitted is the coefficient of the x^0 term.

The CRC polynomial is the remainder resulting from $[x^{16} \times M(x)]$ divided (modulo 2) by G(x):

$$R(x) = x^{16} \times M(x) \mod G(x)$$

6.3 Management frames

A management type frame contains certain mandatory fixed-length fields and some optional variable length components referred to as information elements.

6.3.1 Beacon

A beacon frame contains a Frame Payload that is formatted as shown in Figure 12. It is broadcast by a hub in every beacon period.

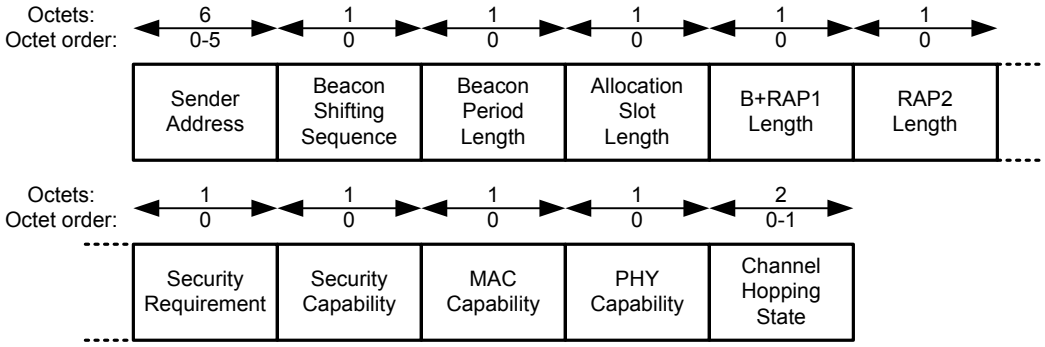


Figure 12 — Frame Payload format for beacon frames

6.3.1.1 Sender Address

The Sender Address field is set to the IEEE MAC address of the hub sending the beacon.

6.3.1.2 Beacon Shifting Sequence

The Beacon Shifting Sequence is formatted as shown in Figure 13 to indicate the beacon transmission time in the current beacon period.

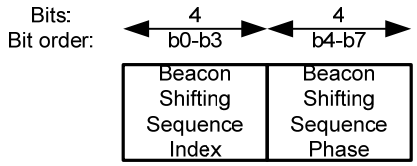


Figure 13 — Beacon Shifting Sequence format

6.3.1.2.1 Beacon Shifting Sequence Index

The Beacon Shifting Sequence Index field is set according to Table 4 to the index *m* of the PN sequence $P_m(n)$ governing the beacon transmission time pattern.

Table 4 — Beacon Shifting sequences

Beacon Shifting Sequence Index decimal value	Beacon Shifting sequence values	Beacon Shifting sequence pattern ("…" denotes pattern repeat)
0	$PN_0(n) = 0, n = 0, 1, \ldots$	$PN_0(n) = 0, 0, 0, 0, \ldots$
1	$PN_1(n) = n \bmod 2, n = 0, 1, \ldots$	$PN_1(n) = 0, 1, 0, 1, \ldots$
2	$PN_2(n) = 2 \times PN_1(n), n = 0, 1, \ldots$	$PN_2(n) = 0, 2, 0, 2, \ldots$
3	$PN_3(n) = n \bmod 4, n = 0, 1, \ldots$	$PN_3(n) = 0, 1, 2, 3, \ldots$
4	$PN_4(n) = [PN_1(n) + PN_3(n)]/2 \bmod 2 + [PN_1(n) + PN_2(n) + PN_3(n)] \bmod 4, n = 0, 1, \ldots$	$PN_4(n) = 0, 1, 3, 2, \ldots$
5	$PN_5(n) = [PN_1(n) + PN_2(n) + PN_3(n)]/2, n = 0, 1, \ldots$	$PN_5(n) = 0, 2, 1, 3, \ldots$
6	$PN_6(n) = \{PN_3(n) + [PN_1(n) + PN_3(n)]/2\} \bmod 4, n = 0, 1, \ldots$	$PN_6(n) = 0, 2, 3, 1, \ldots$
7	$PN_7(n) = PN_2(n) + \{[PN_1(n) + PN_3(n)]/2 \bmod 2\}, n = 0, 1, \ldots$	$PN_7(n) = 0, 3, 1, 2, \ldots$
8	$PN_8(n) = [PN_2(n) + PN_3(n)] \bmod 4, n = 0, 1, \ldots$	$PN_8(n) = 0, 3, 2, 1, \ldots$
9-15	Reserved	Reserved

6.3.1.2.2 Beacon Shifting Sequence Phase

The Beacon Shifting Sequence Phase field is set according to Table 4 to the phase *n* of the chosen PN sequence $P_m(n)$ in the current beacon period.

6.3.1.3 Beacon Period Length

The Beacon Period Length field is set to the length of the current beacon period in units of allocation slots. It is set to 0 to encode a value of 256 allocation slots.

A beacon period must have 4N allocation slots in length, where N is an integer. The allocation slots in a beacon period are numbered 0, 1, …, 4N-1, starting from the allocation slot that starts at the beacon transmission time of the beacon period and to the allocation slot that ends at the end of the beacon period and, if the beacon transmission time is not at the start of the beacon period, back to the allocation slot that starts at the start of the beacon period and finally to the allocation slot that ends at the beacon transmission time, as shown in Figure 14.

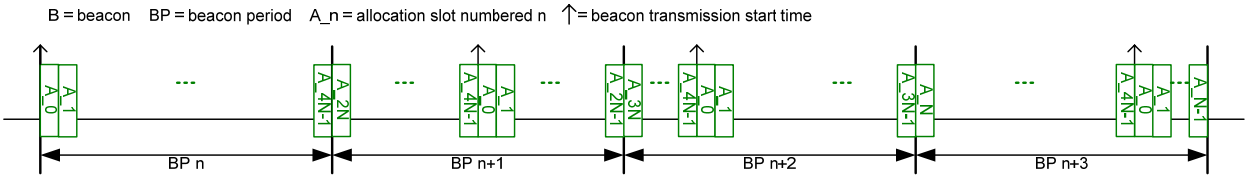


Figure 14 — Allocation slot ordering and numbering in beacon periods

A beacon period has four quarters, which are comprised of allocation slots 0 to N-1, N to 2N-1, 2N to 3N-1, and 3N to 4N-1, respectively. An allocation interval that does not cross a quarter boundary will not be fragmented regardless of which beacon shifting sequence is being used.

6.3.1.4 Allocation Slot Length

The Allocation Slot Length field is set to the length of an allocation slot in units of milliseconds. It is set to 0 to encode a value of 256 milliseconds.

6.3.1.5 B+RAP1 Length

The B+RAP1 Length field is set to the length of the random access phase (RAP) that starts at the end of the beacon frame in the next beacon period, plus the beacon transmission time that precedes. It is set to 0 if no such a RAP is provided. The value of this field must not be smaller than the value of the Minimum B+RAP1 Length field in any Connnection Assignment frame sent by the hub transmitting this beacon.

6.3.1.6 RAP2 Length

The RAP2 Length field is set to the length of the random access phase (RAP) that starts half a beacon period from the start of the beacon frame in the next beacon period. It is set to 0 to if no such a RAP is provided.

6.3.1.7 Security Requirement

The Security Requirement is formatted as shown in Figure 15.

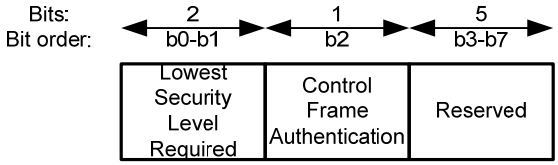


Figure 15 — Security Requirement Format

6.3.1.7.1 Lowest Security Level Required

The Lowest Security Level Required field is set to the lowest security level required by this sender according to Table 5.

Table 5 — Security Requirement field encoding

Field value decimal	Lowest security level required
0	Level 0 – unsecured communication
1	Level 1 – authentication but not encryption
2	Level 2 – authentication and encryption
3-15	Reserved

6.3.1.7.2 Control Frame Authentication

The Control Frame Authentication field is set to 1 if control type frames sent to this sender must be authenticated but not encrypted when they are required to have security level 1 or 2. It is set to 0 if control type frames sent to this sender must be neither authenticated nor encrypted even when they are otherwise required to have security level 1 or 2.

6.3.1.8 Security Capability

The Security Capability is formatted as shown in Figure 16.

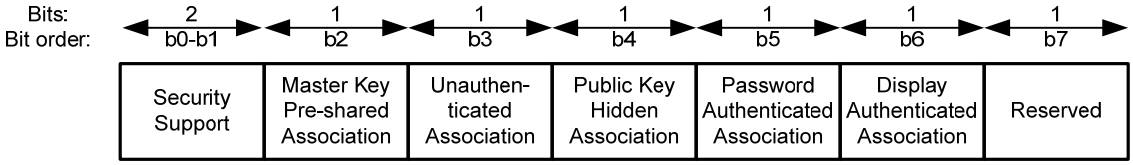


Figure 16 — Security Capability format

6.3.1.8.1 Security Support

The Security Support field is set to the highest security level supported by the sender according to Table 6.

Table 6 — Security Support field encoding

Field value b0 b1	Highest security level supported
00	Level 0 – unsecured communication
10	Level 1 – authentication but not encryption
01	Level 2 – authentication and encryption
11	Reserved

6.3.1.8.2 Master Key (MK) Pre-shared Association

The MK Pre-shared Association field is set to 1 if the sender supports MK pre-shared association, or is set to 0 otherwise.

6.3.1.8.3 Unauthenticated Association

The Unauthenticated Association field is set to 1 if the sender supports unauthenticated association, or is set to 0 otherwise.

6.3.1.8.4 Public Key Hidden Association

The Public Key Hidden Association field is set to 1 if the sender supports public key hidden association, or is set to 0 otherwise.

6.3.1.8.5 Password Authenticated Association

The Password Authenticated Association field is set to 1 if the sender supports password authenticated association, or is set to 0 otherwise.

6.3.1.8.6 Display Authenticated Association

The Display Authenticated Association field is set to 1 if the sender supports display authenticated association, or is set to 0 otherwise.

6.3.1.9 MAC Capability

The MAC Capability is formatted as shown in Figure 17.

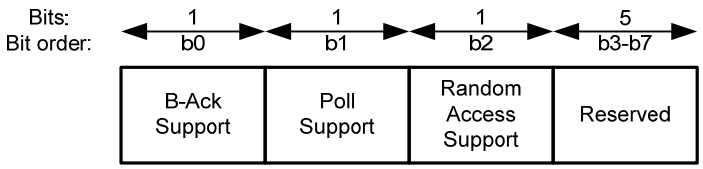


Figure 17 — MAC Capability format

6.3.1.9.1 B-Ack Support

The B-Ack Support field is set to 1 if the sender supports L-Ack and B-Ack acknowledgment policies, or is set to 0 otherwise.

6.3.1.9.2 Poll Support

The Poll Support field is set to 1 if the sender supports polls, or is set to 0 otherwise. A hub supports polls if it can send polls to nodes and participate in the frame transactions sent in its polled allocations. A node supports polls if it can receive polls and initiate frame transactions with the hub in its polled allocations.

6.3.1.9.3 Random Access Support

The Random Access Support field is set to 1 if the sender supports CSMA/CA based random access, or is set to 0 otherwise.

In a beacon containing this field sent out by a hub,

– a value of 1 in this field indicates that the hub will provide a RAP1 with a guaranteed minimum length in each beacon period, and

– a value of 0 indicates that the hub will provide neither RAP1 nor RAP2 in any beacon period.

A hub communicates its guaranteed minimum length for RAP1 to nodes through the Minimum B+RAP1 Length field in its Connection Assignment frames sent to those nodes.

6.3.1.10 PHY Capability

The PHY Capability is formatted as shown in Figure 18

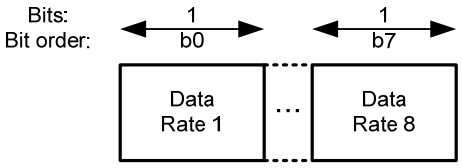


Figure 18 — PHY Capability format

Each Data Rate field is set to the value denoting a data rate supported by the sender according to Table 7.

Table 7 — Data Rate field encoding

Field value	Data rate

6.3.1.11 Channel Hopping State

The Channel Hopping State field is set to the current state of a 16-bit maximum-length linear feedback shift register (LFSR) used to generate the channel hopping sequence by the hub sending this beacon.

The Channel Hopping State field is set to 0 if the hop is to dwell on the current channel for an indefinite period of time (i.e., without channel hopping).

6.3.2 Association

An Association frame contains a Frame Payload that is formatted as shown in Figure 19. It is exchanged between a node and a hub to activate a pre-shared MK or generate a new shared MK.

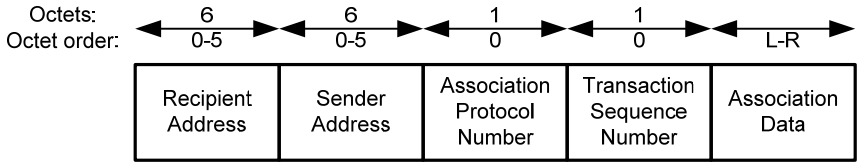


Figure 19 — Frame Payload format for Association frames

6.3.2.1 Recipient Address

The Recipient Address field is set to the IEEE MAC address of the recipient of the current frame.

6.3.2.2 Sender Address

The Sending Address field is set to the IEEE MAC address of the sender of the current frame.

6.3.2.3 Association Protocol Number (APN)

The Association Protocol Number field is set according to Table 8 to indicate an association protocol being used for the association.

Table 8 — Association Protocol Number field encoding

Field value decimal	Association protocol
0	Master key pre-shared association
1	Unauthenticated association
2	Public key hidden association
3	Password authenticated association
4	Display authenticated association
5-255	Reserved

6.3.2.4 Transaction Sequence Number

The Transaction Sequence Number field is set to the number (i.e., position) of the current Association frame in the run of the chosen association protocol. In particular, it is set to 1 in the first Association frame of the protocol, 2 in the second Association frame, 3 in the third, etc. The first Association frame is the Association frame transmitted or retransmitted by the node initializing the association, the second Association frame is the Association frame transmitted or retransmitted by the responding hub. The other values of the field are reserved.

6.3.2.5 Association Data

The Association Data field is specific to the association protocol being used.

For master key pre-shared association, the Association Data field is not present.

For unauthenticated association, public key hidden association, password authenticated association, and display authenticated association, the Association Data field is formatted as shown in Figure 20.

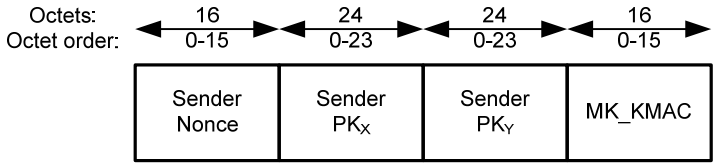


Figure 20 — Association Data format for association protocols 1-4

6.3.2.5.1 Sender Nonce

The Sender Nonce field is set to a statistically unique number per sender and per association procedure, except otherwise indicated:

- For unauthenticated association, public key hidden association, and password authenticated association,
  - in the first and second Association frames of the current association procedure, the field is set afresh and independently to an integer randomly drawn with a uniform distribution over the interval (0, 2^128);
  - in the third Association frame, the field is set to its value contained in the first Association frame of the procedure.
- For display authenticated association,
  - in the first Association frame of the current association procedure, the field is set to 0;

- in the second and third Association frames, the field is set afresh and independently to an integer randomly drawn with a uniform distribution over the interval (0, 2^128).

6.3.2.5.2 Sender PK_x

The Sender PK_x field is set to the *X*-coordinate of the sender's elliptic curve public key, except otherwise indicated:

- For unauthenticated association,
  - in the first and second Association frames of the current association procedure, the field is set to the *X*-coordinate of the sender's elliptic curve public key $PK = (PK_x, PK_Y)$;
  - in the third Association frame, the field is set to its value contained in the first Association frame of the procedure.
- For public key hidden association,
  - in the first and third Association frames of the current association procedure, the field is set to 0;
  - in the second Association frame, the field is set to the X-coordinate of the sender's elliptic curve public key $PK = (PK_x, PK_Y)$.
- For password authenticated association,
  - in the first and third Association frames of the current association procedure, the field is set to the *X*-coordinate of the sender's password-scrambled elliptic curve public key $PK' = (PK'_x, PK'_Y)$;
  - in the second Association frame, the field is set to the *X*-coordinate of the sender's elliptic curve public key $PK = (PK_x, PK_Y)$.
- For display authenticated association,
  - in the first Association frame of the current association procedure, the field is set to 0;
  - in the second and third Association frames, the field is set to the *X*-coordinate of the sender's elliptic curve public key $PK = (PK_x, PK_Y)$.

6.3.2.5.3 Sender PK_Y

The Sender PK_Y field is set to the *Y*-coordinate of the sender's elliptic curve public key, except otherwise indicated:

- For unauthenticated association,
  - in the first and second Association frames of the current association procedure, the field is set to the *Y*-coordinate of the sender's elliptic curve public key $PK = (PK_x, PK_Y)$;
  - in the third Association frame, the field is set to its value contained in the first Association frame of the procedure.
- For public key hidden association,
  - in the first and third Association frames of the current association procedure, the field is set to 0;
  - in the second Association frame, the field is set to the *Y*-coordinate of the sender's elliptic curve public key $PK = (PK_x, PK_Y)$.
- For password authenticated association,
  - in the first and third Association frames of the current association procedure, the field is set to the *Y*-coordinate of the sender's password-scrambled elliptic curve public key $PK' = (PK'_x, PK'_Y)$;
  - in the second Association frame, the field is set to the *Y*-coordinate of the sender's elliptic curve public key $PK = (PK_x, PK_Y)$.
- For display authenticated association,

- in the first Association frame of the current association procedure, the field is set to 0;
- in the second and third Association frames, the field is set to the *Y*-coordinate of the sender's elliptic curve public key $PK = (PK_x, PK_Y)$.

6.3.2.5.4 MK_KMAC

The MK_KMAC field is set to a keyed message authentication check (KMAC) for certain fields of the frame payloads of the Association frames of the current association procedure, except otherwise indicated:

- For unauthenticated association,
  - in the first Association frame of the current association procedure, the field is set to 0;
  - in the second Association frame, the field is set to a KMAC for certain fields of the frame payloads of the first and second Association frames of the procedure;
  - in the third Association frame, the field is set to a KMAC for certain fields of the frame payloads of the second and third Association frames.
- For public key hidden association,
  - in the first Association frame of the current association procedure, the field is set to 0;
  - in the second Association frame, the field is set to a KMAC for certain fields of the frame payloads of the first and second Association frames of the procedure if the sender of this frame has the public key of the sender of the first Association frame, or it is set to 0 otherwise;
  - in the third Association frame, the field is set to a KMAC for certain fields of the frame payloads of the second and third Association frames.
- For password authenticated association,
  - in the first Association frame of the current association procedure, the field is set to 0;
  - in the second Association frame, the field is set to a KMAC for certain fields of the frame payloads of the first and second Association frames of the procedure if the sender of this frame has a shared password with the sender of the first Association frame, or it is set to 0 otherwise;
  - in the third Association frame, the field is set to a KMAC for certain fields of the frame payloads of the second and third Association frames.
- For display authenticated association,
  - in the first Association frame of the current association procedure, the field is set to an initial commitment equal to a KMAC for certain fields of the frame payload of the third Association frame of the procedure;
  - in the second and third Association frames, the field is set to 0.

6.3.3 Disassociation

A Disassociation frame contains a Frame Payload that is formatted as shown in Figure 21. It is transmitted by either an associated node or a hub to repeal an existing association, i.e., the shared master key (MK).

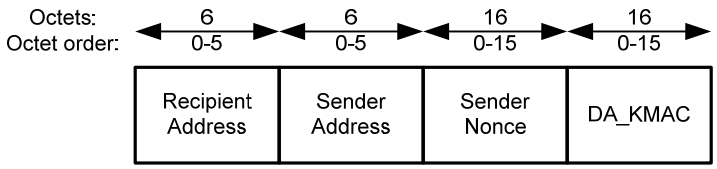


Figure 21 — Frame Payload format for Disassociation frames

6.3.3.1 Recipient Address

The Recipient Address field is set to the IEEE MAC address of the recipient of the current frame.

6.3.3.2 Sender Address

The Sending Address field is set to the IEEE MAC address of the sender of the current frame.

6.3.3.2.1 Sender Nonce

The Sender Nonce field is set to a statistically unique number per sender and per disassociation procedure, i.e., it is set afresh and independently to an integer randomly drawn with a uniform distribution over the interval (0, 2^128).

6.3.3.2.2 DA_KMAC

The DA_KMAC field is set to a keyed message authentication check (KMAC) for certain fields of the frame payload of this Disassociation frame.

6.3.4 Pairwise Temporal Key (PTK)

A PTK frame contains a Frame Payload that is formatted as shown in Figure 22. It is exchanged between a node and the hub with which the node is associated to create a PTK based on a shared master key (MK).

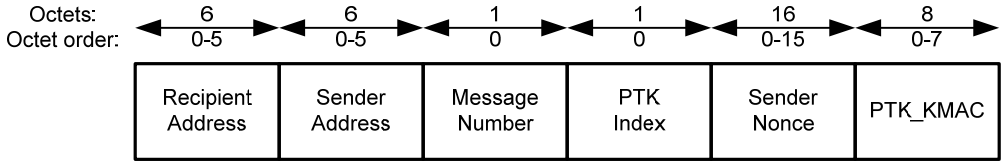


Figure 22 — Frame Payload format for PTK frames

6.3.4.1 Recipient Address

The Recipient Address field is set to the IEEE MAC address of the recipient of the current frame.

6.3.4.2 Sender Address

The Sending Address field is set to the IEEE MAC address of the sender of the current frame.

6.3.4.3 Message Number

The Message Number field is set to the number (i.e., position) of the current PTK frame in the current PTK creation procedure. In particular, it is set to 1 in the first PTK frame of the procedure, 2 in the second PTK frame, and 3 in the third. The first PTK frame is the PTK frame transmitted or retransmitted by the node or hub initializing the procedure, the second PTK frame is the PTK frame transmitted or retransmitted by the responding hub or node, and the third PTK frame is the PTK frame transmitted or retransmitted by the initiating node or hub again. The other values of the field are reserved.

6.3.4.4 PTK Index

The PTK Index field in the first PTK frame transmitted or retransmitted by the node initiating the PTK creation procedure is set as follows to identify the PTK being created:

– If no PTK was previously created with the responding node, it is set to 0.

– Otherwise, it is set to 1 minus its value used in successfully creating the last PTK between the two nodes.

The PTK Index field in the second and third PTK frames of the PTK creation procedure is set to its value contained in the first PTK frame of the procedure.

The PTK Index field takes on a value of either 0 or 1. The other values of the field are reserved.

6.3.4.5 Sender Nonce

The Sender Nonce field is set to a statistically unique number per sender and per PTK creation procedure:

– In the first and second PTK frames of the current PTK creation procedure, it is set afresh and independently to an integer randomly drawn with a uniform distribution over the interval (0, 2^128).

– In the third PTK frame of the procedure, it is set to its value contained in the first PTK frame.

6.3.4.6 PTK_KMAC

The PTK_KMAC field is set to a keyed message authentication check (KMAC) certain fields of the frame payloads of the PTK frames of the current PTK creation procedure, except otherwise indicated:

– In the first PTK frame of the current PTK creation procedure, it is set to 0.

– In the second PTK frame, it is set to a KMAC for certain fields of the frame payloads of the first and second PTK frames of the procedure if the sender of this frame has a shared master key (MK) with the sender of the first PTK frame, or it is set to 0 otherwise.

– In the third PTK frame, it is set to a KMAC for certain fields of the frame payloads of the second and third PTK frames.

6.3.5 Group Temporal Key (GTK)

A GTK frame contains a Frame Payload that is formatted as shown in Figure 23. It is transmitted by a hub to distribute a GTK to an associated node for securing broadcast or multicast traffic.

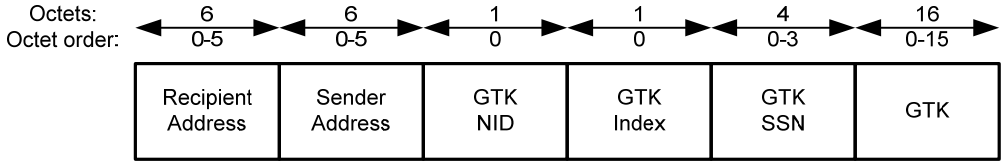


Figure 23 — Frame Payload format for GTK frames

6.3.5.1 Recipient Address

The Recipient Address field is set to the IEEE MAC address of the recipient of the current frame.

6.3.5.2 Sender Address

The Sending Address field is set to the IEEE MAC address of the sender of the current frame.

6.3.5.3 GTK NID

The GTK NID field is set according to Table 11 to the broadcast or multicast NID that is to appear in the NID field of the MAC header of the frames secured by the GTK distributed in this frame.

6.3.5.4 GTK Index

The GTK Index field is set as follows to identify the GTK being distributed:

– If no GTK was previously distributed by this hub for the GTK NID indicated in this frame, it is set to 0.

– Otherwise, it is set to 1 minus its value used in successfully distributing the last GTK by this hub for the indicated GTK NID.

The GTK Index field takes on a value of either 0 or 1. The other values of the field are reserved.

6.3.5.5 GTK SSN

The GTK SSN field is set to the security sequence number of the last frame secured with the GTK distributed in this frame and addressed to the GTK NID indicated in this frame.

6.3.5.6 GTK

The GTK field is set to the bit string representing the GTK being distributed in this frame.

6.3.6 Connection Request

A Connection Request frame contains a Frame Payload that is formatted as shown in Figure 24. It is transmitted by a node to request creation or modification of a connection, in particular, a Connected_NID, a power management profile, and an access allocation contract, with a hub.

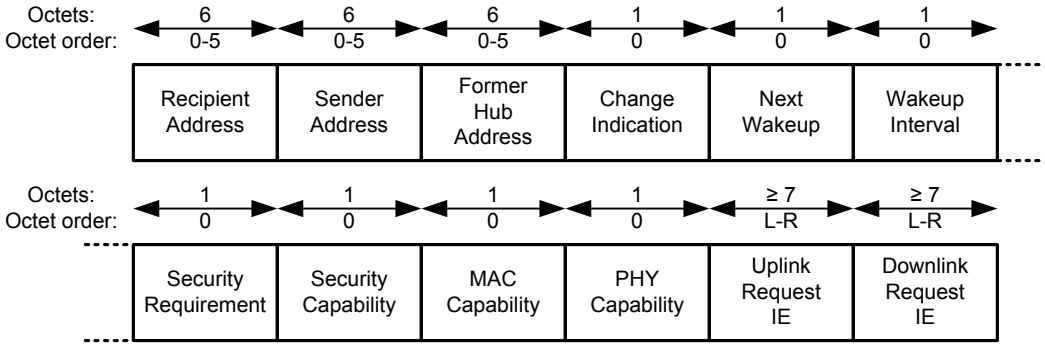


Figure 24 — Frame Payload format for Connection Request frames

6.3.6.1 Recipient Address

The Recipient Address field is set to the IEEE MAC address of the recipient of the current frame.

6.3.6.2 Sender Address

The Sending Address field is set to the IEEE MAC address of the sender of the current frame.

6.3.6.3 Former Hub Address

The Former Hub Address field is set as follows:

– If the sending node was not connected to another hub previously, it is set to 0.

– Otherwise, it is set to the IEEE MAC address of the hub with which the node was last connected.

6.3.6.4 Change Indication

The Change Indication is formatted as shown in Figure 25. A bit field is set to 1 if the field it denotes has been changed in this connection request since the last one, or it is set to 0 otherwise.

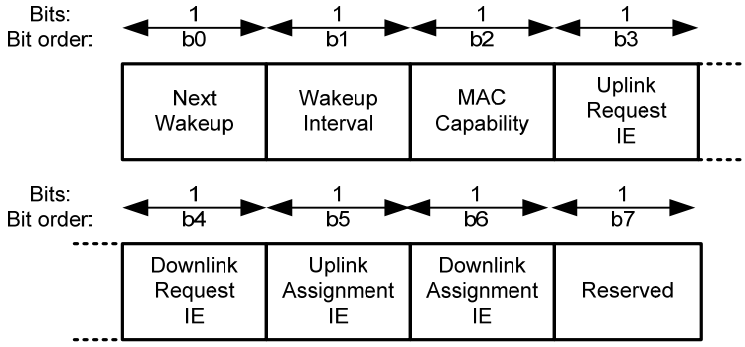


Figure 25 — Change Indication format

6.3.6.5 Next Wakeup

The Next Wakeup field is set to the sequence number of the beacon transmitted in the beacon period, referred to as the node's wakeup beacon period, in which the node will wake up for frame reception and transmission.

6.3.6.6 Wakeup Interval

The Wakeup Interval field is set to the length, in units of beacon periods, between the start of successive wakeups of this node, and is effective from the next wakeup indicated in the preceding field. It is set to 0 to encode a value of 256 beacon periods.

The value of this field determines whether the IEs in this frame and the responding Connection Assignment frame denote 1-periodic or m-periodic allocations:

– If Wakeup Interval = 1, these IEs denote 1-periodic allocations.

– If Wakeup Interval ≠ 1, these IEs denote m-periodic allocations.

6.3.6.7 Security Requirement

The Security Requirement field is formatted as defined in 6.3.1.7.

6.3.6.8 Security Capability

The Security Capability is formatted as defined in 6.3.1.8.

6.3.6.9 MAC Capability

The MAC Capability is formatted as defined in 6.3.1.9.

6.3.6.10 PHY Capability

The PHY Capability is formatted as defined in 6.3.1.9.2.

6.3.6.11 Uplink Request IE

The Uplink Request IE is defined in 6.6.1.

6.3.6.12 Downlink Request IE

The Uplink Request IE is defined in 6.6.2.

6.3.7 Connection Assignment

A Connection Assignment frame contains a Frame Payload that is formatted as shown in Figure 26. It is transmitted by a hub to respond to a connection request or to change an earlier connection assignment.

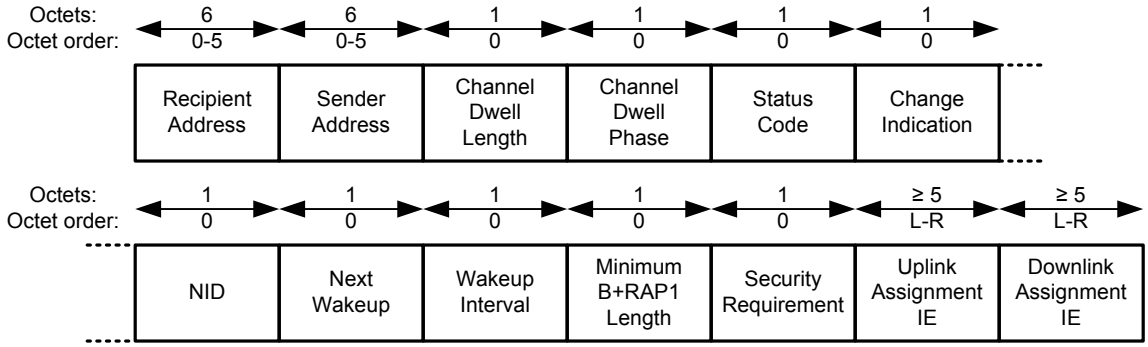


Figure 26 — Frame Payload format for Connection Assignment frames

6.3.7.1 Recipient Address

The Recipient Address field is set to the IEEE MAC address of the recipient of the current frame.

6.3.7.2 Sender Address

The Sending Address field is set to the IEEE MAC address of the sender of the current frame.

6.3.7.3 Channel Dwell Length

The Channel Dwell Length field is set to the length of the time, in units of beacon periods, over which the hub sending this frame is to operate at a chosen channel before hopping to another one. It is set to 0 to encode a value of 256 beacon periods.

6.3.7.4 Channel Dwell Phase

The Channel Dwell Phase field is set to H if the hub sending the current beacon frame is to hop to the next channel in its channel hopping sequence at the end of the next Hth beacon period not counting the current beacon period if $H > 0$ or at the end of the current beacon period if $H = 0$.

6.3.7.5 Status Code

The Status Code field is set to the status of the connection assignment encoded according to Table 9.

Table 9 — Status Code field encoding

Field value decimal	Status
0	Connection request accepted
1	Connection request rejected – no more channel bandwidth for a new connection
2	Connection request rejected – no more Connected_NID for a new connection
3	Connection request rejected – no more internal resources for a new connection
4	Connection rejected – the highest security level supported by the requester not high enough
5	Connection rejected – the lowest security level required by the requester is higher than the highest security level supported by the responder
6	Connection request rejected – no reason
7	Connection assignment modified
8-255	Reserved

6.3.7.6 Change Indication

The Change Indication is formatted as shown in Figure 25. A bit field is set to 1 if the field it denotes has been changed in this connection assignment since last connection request or connection assignment, or it is set to 0 otherwise.

6.3.7.7 NID

The NID field is set to a NID that is uniquely assigned or reassigned to the addressed node within the subnet according to Table 11.

6.3.7.8 Next Wakeup

The Next Wakeup field is defined as in 6.3.6.5, except that it is set by the hub sending this frame for the addressed node. The field in this frame supersedes the field in the last Connection Request or Connection Assignment frame.

6.3.7.9 Wakeup Interval

The Wakeup Interval field is defined as in 6.3.6.6, except that it is set by the hub sending this frame for the addressed node. The field in this frame supersedes the field in the last Connection Request or Connection Assignment frame.

6.3.7.10 Minimum B+RAP1 Length

The Minimum B+RAP1 Length field is set to the smallest value guaranteed for the B+RAP1 Length field in the beacons from the hub sending this frame, if the Random Access Support field of the MAC Capability of the beacons is set to 1. It is set to 0 otherwise.

6.3.7.11 Security Requirement

The Security Requirement field is formatted as defined in 6.3.1.7.

6.3.7.12 Uplink Assignment IE

The Uplink Assignment IE is defined in 6.6.3.

6.3.7.13 Downlink Assignment IE

The Downlink Assignment IE is defined in 6.6.4.

6.3.8 Disconnection

A Disconnection frame contains a Frame Payload that is formatted as shown in Figure 27. It is transmitted by a hub to repeal the connection with a node or by a node to repeal the connection with a hub.

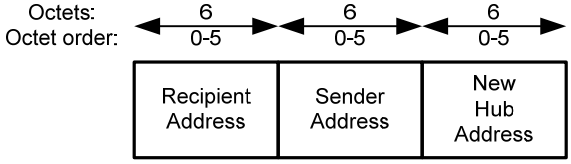


Figure 27 — Frame Payload format for Disconnection frames

6.3.8.1 Recipient Address

The Recipient Address field is set to the IEEE MAC address of the recipient of the current frame.

6.3.8.2 Sender Address

The Sending Address field is set to the IEEE MAC address of the sender of the current frame.

6.3.8.3 New Hub Address

The New Hub Address field in Disconnection frames sent by a node is set as follows:

– If the node is not newly connected with another hub, it is set to 0.

– Otherwise, it is set to the MAC address of the hub with which the node is newly connected.

The New hub Address field in Disconnection frames sent by a hub is reserved.

6.4 Control frames

A control type frame contains no frame payload or a frame payload of variable length.

6.4.1 Immediate Acknowledgement (I-Ack)

An I-Ack frame contains no Frame Payload. It is transmitted by a node or a hub to acknowledge receipt of the preceding frame.

6.4.2 Block Acknowledgement (B-ACK)

A B-Ack frame contains a Frame Payload that is formatted as shown in Figure 28. It is transmitted by a node or a hub to acknowledge the reception status of certain preceding data type frames each containing a whole MSDU.

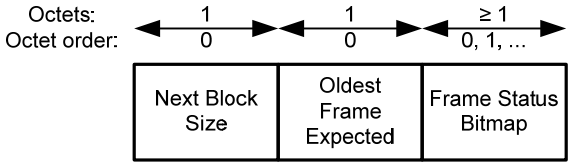


Figure 28 — Frame Payload format for B-Ack frames

6.4.2.1 Next Block Size

The Next Block Size field is set to the maximum number of data type frames permitted in the next block transmission from the acknowledged node or hub to the acknowledging hub or node, where the next block transmission is a transmission of data type frames whose reception status will be provided in the next B-Ack frame and whose frame subtype is the same as that of the data type frame preceding this B-Ack frame.

6.4.2.2 Oldest Frame Expected

The Oldest Frame Expected field is set to the sequence number of the oldest frame that is of the same frame subtype as the frame preceding this B-Ack frame and that is still expected to be, but not yet, received.

6.4.2.3 Frame Status Bitmap

The Frame Status Bitmap field is set as follows to indicate the reception status of the frames that are of the same frame subtype as the frame preceding this B-Ack frame and that are newer than the frame indicated in the Oldest Frame Expected field:

– The field comprises $\lceil N/8 \rceil$ octets, where N is the number of these frames and $\lceil f \rceil$ is the least integer that is not smaller than f.

– The least-significant bit of these octets denotes the oldest of these frames.

– Each successive bit denotes a successive frame, i.e., a frame with a successive sequence number, of the same frame subtype.

– A bit is set to 1 if the corresponding frame is received.

6.4.3 Immediate Acknowledgement + Poll (I-Ack+Poll)

An I-Ack+Poll frame contains no Frame Payload. It is transmitted by a hub to acknowledge receipt of the preceding frame and to send a poll to the addressed node. The I-Ack+Poll frame is equivalent in function to an I-Ack frame followed by a Poll frame.

6.4.4 Block Acknowledgement +Poll (B-ACK+Poll)

A B-Ack+Poll frame contains a Frame Payload that is formatted as defined in 6.4.2. It is transmitted by a hub to acknowledge the reception status of certain preceding data type frames and to send a poll to the addressed node. The B-Ack+Poll frame is equivalent in function to a B-Ack frame followed by a Poll frame.

6.4.5 Poll

A Poll frame contains no Frame Payload. It is transmitted by a hub to grant to the addressed node an immediate polled allocation that starts TIFS after the end of the frame or to inform the node of a future poll.

6.5 Data frames

A data type frame contains a nonzero-length Frame Payload. It contains a full or fragmented MSDU, and is transmitted by a hub to a connected node or by a node to the hub with which the node is connected.

A data type frame has a user-defined frame subtype that is set as follows: All data type frames of the same frame subtype carry MSDUs of the same application, or a group of MSDUs of the same attributes.

6.6 Information elements (IEs)

An IE is formatted as shown in Figure 29. It is contained in certain management type frames.

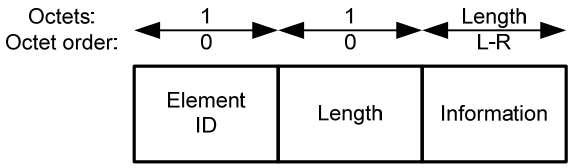


Figure 29 — IE format – general

The Element ID field is set to the value that identifies the information element according to Table 10.

The Length field is set to the length, in octets, of the IE-specific Information field that follows.

The Information field is set based on the Element ID as defined below.

Table 10 — Information elements defined in this standard

Element ID decimal value	IE name	Description
0	Reserved	Reserved
1	Uplink Request IE	Specifies scheduled uplink allocation requirements by a node
2	Downlink Request IE	Specifies scheduled downlink allocation requirements by a node
3	Uplink Assignment IE	Specifies scheduled uplink allocation assignments to a node
4	Downlink Assignment IE	Specifies scheduled downlink allocation assignments to a node
5-244	Reserved	Reserved
255	Application Specific IE	Provides user-defined application-specific information

6.6.1 Uplink Request IE

An Uplink Request IE is formatted as shown in Figure 30. It is contained in Connect Request frames to indicate the request for creation or modification of some scheduled uplink allocations of the sender.

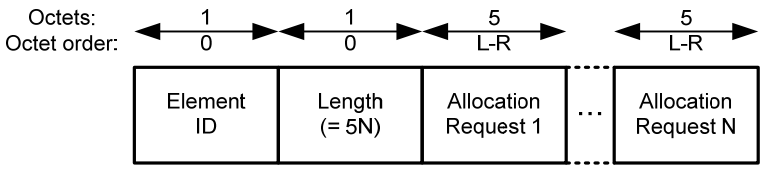


Figure 30 — Uplink Request IE format

Each Allocation Request is formatted as shown in Figure 31 to describe the requirements of an allocation for the data belonging to a given user priority.

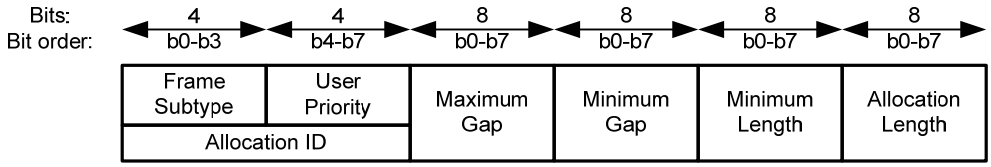


Figure 31 — Allocation Request format

6.6.1.1 Allocation ID

The Allocation ID identifies an allocation requested by the node. It is comprised of the Frame Subtype and User Priority fields as defined below.

6.6.1.2 Frame Subtype

The Frame Subtype field is set to the frame subtype of the data type frames to be transferred in this requested allocation.

6.6.1.3 User Priority

The User Priority field is set to the user priority of the frame payloads to be transferred in this requested allocation.

6.6.1.4 Maximum Gap

The Maximum Gap field is set to the largest length, in units of allocation slots, of the gap between any two adjacent allocation intervals of this requested allocation in the same beacon period or across beacon periods, if the Wakeup Interval field in the same frame has a value of 1. It is reserved otherwise.

6.6.1.5 Minimum Gap

The Minimum Gap field is set to the smallest length, in units of allocation slots, of the gap between any two adjacent allocation intervals of this requested allocation in the same beacon period or across beacon periods.

6.6.1.6 Minimum Length

The Minimum Length field is set to the smallest length, in units of allocation slots, of any of the allocation intervals of this requested allocation.

6.6.1.7 Allocation Length

The Allocation Length field is set to the overall length, in units of allocation slots, of the allocation intervals of this requested allocation in a wakeup beacon period.

6.6.2 Downlink Request IE

A Downlink Request IE is formatted as shown in Figure 30 in conjunction with Figure 31. It is contained in Connect Request frames to indicate the request for creation or modification of some scheduled downlink allocations of the sender.

6.6.3 Uplink Assignment IE

An Uplink Assignment IE is formatted as shown in Figure 32. It is contained in Connection Assignment frames to indicate the assignment of some scheduled uplink allocations to the addressed node.

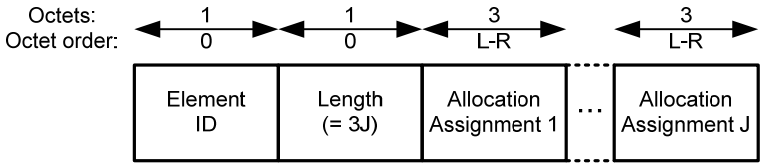


Figure 32 — Uplink Assignment IE format

Each Allocation Assignment is formatted as shown in Figure 33 to specify an allocation interval of an assigned allocation for the data belonging to a given user priority. Multiple Allocation Assignment fields are included in this IE to specify an allocation comprising multiple allocation intervals.

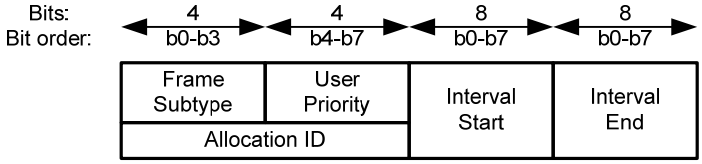


Figure 33 — Allocation Assignment format

6.6.3.1 Allocation ID

The Allocation ID identifies an allocation assigned to the node. It is set to the Allocation ID used to identify the same allocation requested earlier by the node.

6.6.3.2 Frame Subtype

The Frame Subtype field is set to the frame subtype of the data type frames to be transferred in this assigned allocation.

6.6.3.3 User Priority

The User Priority field is set to the user priority of the frame payloads to be transferred in this assigned allocation.

6.6.3.4 Interval Start

The Interval Start field is set to the number of the allocation slot that starts an allocation interval of this assigned allocation.

6.6.3.5 Interval End

The Interval End field is set to the number of the allocation slot that ends the allocation interval whose start time is specified in the preceding field.

6.6.4 Downlink Assignment IE

An Uplink Assignment IE is formatted as shown in Figure 32 in conjunction with Figure 33. It is contained in Connection Assignment frames to indicate the assignment of some scheduled downlink allocations to the addressed node.

6.6.5 Application Specific IE

An Application Specific IE is formatted as shown in Figure 34. It is contained in some management type frames to convey application-specific information.

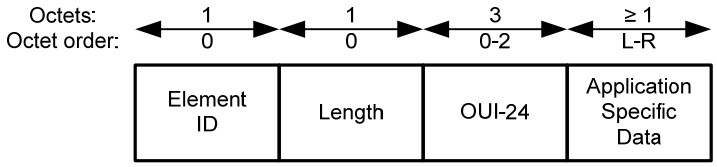


Figure 34 — Application Specific IE format

6.6.5.1 OUI-24

The OUI-24l field is set to the 24-bit Organizationally Unique Identifier (OUI) assigned by the IEEE Registration Authority to the vendor or manufacturer that defines this IE.

6.6.5.2 Application Specific Data

The Application Specific Data field is set by the owner ("assignee" in IEEE terms) of the OUI.

7 MAC functions

This clause specifies MAC sublayer functionality. The rules for preparing frame transmission and processing frame reception are given in 7.1. Medium access methods are described in 7.2-7.6. Clock synchronization and guartime provisioning are specified in 7.7. Power management is provided in 7.8. Channel change mechanisms are presented in 7.9. Multi-rate support is described in 7.10. Application Specific IE usage is clarified in 7.11. MAC sublayer parameters are given in 7.12.

7.1 Frame processing

This subclause provides rules on preparing MAC frames for transmission and processing them on reception.

7.1.1 Abbreviated addressing

A two-octet hub identifier (HID) selected from an integer between x0000 and xFFFF shall be used as a hub's abbreviated address contained in the MAC header of the frames sent from or to the hub.

A one-octet node identifier (NID) selected in accordance with Table 11 shall be used as a node's abbreviated address contained in the MAC header of the frames sent from or to the node, or shall be used as a set of nodes' common abbreviated address contained in the MAC header of the frames sent to the nodes by a hub.

Table 11 — NID selection

NID value in hex	NID subtotal	NID notation	NID usage
x00	1	Unconnected_Broadcast_NID	For broadcast to unconnected nodes
x01-0F	15	Unconnected_NID	For unicast to unconnected nodes
x10-EF	224	Connected_NID	For unicast to/from connected nodes
xF0-FE	15	Multicast_NID	For multicast to connected nodes
xFF	1	Broadcast_NID	For broadcast to all nodes

A node shall choose as its NID one of the Unconnected_NID values in its first frame sent to a hub, and retries thereof when applicable. The node should not use an NID that is being used by another node in the same subnet.

The hub should use this Unconnected_NID if the latter is not being used for another unconnected node in the same subnet, or otherwise should choose and use a different Unconnected_NID that is not being used for another unconnected node in the subnet, in its management type frames sent to the node, until the node is connected with the hub. A node is connected with the hub when it receives a status 0 Connection Assignment frame from the hub. A status 0 Connection Assignment frame is one with the Status Code field set to 0 indicating "connection request accepted".

An unconnected node shall be ready to receive management type frames from the hub to which it has sent its management type frames, without regard to the NID field of the MAC header. Once it receives from the hub a management type frame with the NID field set to its current Unconnected_NID but with the Recipient Address field not set to its own IEEE MAC address, or after it acknowledges such a received frame, it shall no longer acknowledge frames addressed to that Unconnected_NID, and shall choose another Unconnected_NID as its NID.

The hub shall choose a Connected_NID that is not being used for another node in the subnet, and shall use that Connected_NID in a status 0 Connection Assignment frames and all subsequent frames sent to the node.

The hub shall not change the node's NID in its control type frames sent to the node.

The node shall use the NID contained in the last frame received from the hub in its subsequent frames sent to the hub, if that frame contained a Recipient Address field set to its IEEE MAC address.

7.1.2 Full addressing

A separate IEEE MAC address shall be used to uniquely identify a sender or a recipient when desired. In particular, the IEEE MAC address of a hub sending a beacon is included in the MAC frame body of the beacon. The IEEE MAC address of the sender and that of the recipient of other management type frames are included in the MAC frame body of those frames.

Exceptionally, the Recipient Address field of the first management type frame sent by an unconnected node to a hub may be set as follows:

– Its 16 least-significant bits are set to the HID of the hub, i.e., the 16 least-significant bits of the IEEE MAC address of the hub.

– Its 32 most-significant bits are set to 0.

A recipient shall check the Recipient Address and Sender Address fields of the MAC frame body of a received management type frame to determine if the frame was indeed addressed to it from an expected sender, taking the aforementioned exception into account.

7.1.3 Frame reception

A node shall receive a frame if the following conditions are met:

– The HID field of the MAC header of the frame is set to the HID of the desired hub to exchange frames with.

– The NID field of the MAC header of the frame is set to its own NID, a Multicast_NID it has subscribed to, or the Broadcast_NID.

– The Protocol Version of the MAC header of the frame is set to a value it supports.

– The FCS of the frame is valid, i.e., equal to the FCS value it calculates over the applicable fields received.

A hub shall receive a frame if the following conditions are met:

– The HID field of the MAC header of the frame is set to its own HID.

– The NID field of the MAC header of the frame is set to the NID of an expected node or an Unconnected_NID.

– The Protocol Version of the MAC header of the frame is set to a value it supports.

– The FCS of the frame is valid.

The node or the hub shall ignore a received frame, aside from performing applicable acknowledgment, whose MAC frame body has a Sender Address field that is not set to the IEEE MAC address of the expected sender or whose MAC frame body has a Recipient Address field that is not set to its own IEEE MAC address.

The node or the hub shall ignore a received frame, aside from performing applicable acknowledgment, that is detected to be a duplicate as described in 7.1.7.

7.1.4 Frame transfer

A sender shall transmit MSDUs to be carried in data type frames of the same frame subtype and addressed to the same recipient(s) in the octet order in which they arrived at the local MAC SAP.

When fragmenting an MSDU, the sender shall extract the first fragment, the second fragment, and so on, in sequential octet order. The sender shall transmit the first fragment, then the second fragment, and so on, accordingly.

A sender may transmit an MSDU earlier than another MSDU, even if the former arrived at the local MAC SAP later than the latter, so long as the two MSDUs are carried in data type frames not of the same frame subtype or not addressed to the same recipient(s).

A recipient shall release to the MAC client MSDUs that were transmitted by the sender and carried in data type frames of the same frame subtype in the octet order in which they were received.

7.1.5 Frame retry

A node or a hub may retry a frame, i.e., may transmit a frame that was previously transmitted but not necessarily received, to the same recipient(s), as appropriate, taking into consideration such factors as delay requirements, fairness policies, channel conditions, and medium availability.

7.1.6 Frame acknowledgement

A sender shall set the Ack Policy field of the MAC header of a frame to be transmitted according to Table 2 and Table 12. A recipient shall acknowledge a received frame if the criteria in 7.1.3 for qualifying a frame as received are met and if required by the acknowledgment policy set in the frame as further described below. The use of various acknowledgment policies is illustrated in Figure 37.

Table 12 — Acknowledgment (Ack) Policy field setting

Frame Type name	Frame Subtype name	Ack Policy field
Management	Beacon	N-Ack
Management	Association	I-Ack
Management	Disassociation	I-Ack
Management	PTK	I-Ack
Management	GTK	I-Ack
Management	Connection Request	I-Ack
Management	Connection Assignment	I-Ack
Management	Disconnection	I-Ack
Control	I-Ack	N-Ack
Control	B-Ack	N-Ack
Control	I-Ack+Poll	N-Ack
Control	B-Ack+Poll	N-Ack
Control	Poll	N-Ack
Data	User defined data subtype	N-Ack, I-Ack, L-Ack, or B-Ack

7.1.6.1 No Acknowledgment (N-Ack)

A recipient shall not acknowledge a received frame containing an Ack Policy field set to N-Ack, either immediately or later.

A sender may retry a data type frame containing an Ack Policy field set to N-Ack, as appropriate.

7.1.6.2 Immediate Acknowledgment (I-Ack)

A recipient shall acknowledge a received frame containing an Ack Policy field set to I-Ack by sending an I-Ack frame back TIFS after the end of the received frame.

7.1.6.3 Block Acknowledgment Later (L-Ack) and Block Acknowledgment (B-Ack)

A source—a node or a hub—may send a frame with the Ack Policy field set to B-Ack if the following two conditions are satisfied:

– The frame contains a whole MSDU.

– The recipient supports B-Ack as indicated in the latter's MAC Capability field.

The recipient shall acknowledge a received frame with the Ack Policy field set to B-Ack by sending a B-Ack frame back TIFS after the end of the received frame. The B-Ack frame shall contain a frame payload as defined in 6.4.2 unless the following two conditions are both true:

– No older frames of the same frame subtype as the last received frame are still expected to be received.

– Only one frame in the next block transmission is allowed.

The source node may send a frame with the Ack Policy field set to L-Ack if the following conditions are all satisfied:

– The frame contains a whole MSDU.

– It has sent a frame of the same frame subtype with the Ack Policy field set to B-Ack and received a B-Ack frame acknowledging that frame and containing a frame payload.

– The B-Ack frame was the last B-Ack frame received from the recipient.

The source shall not transmit more frames in a block transmission than allowed as specified in the last B-Ack frame received. The source shall end a block transmission with a frame containing an Ack Policy field set to B-Ack.

The source shall send frames in a block transmission in the order of increasing sequence number values, which are not necessarily consecutive if the block transmission contains retransmitted frames, considering that sequence number wraparound is also increasing the sequence number value. The source shall not retransmit frames that are older than the frame indicated in the Oldest Frame Expected field of the last B-Ack frame received. It should retransmit frames, starting with the oldest frame expected, that were not received as indicated in the Frame Status Bitmap field of that B-Ack frame.

The source, once starting a block transmission, shall not transmit frames of another frame type or subtype until it has finished the block transmission.

The recipient shall not acknowledge immediately a received frame containing an Ack Policy field set to L-Ack. Rather, it shall indicate the reception status of the frames newer than the oldest frame still expected through the B-Ack frame it returns at the end of the current block transmission.

The source may retransmit in an appropriate time the last frame which had the Ack Policy field set to B-Ack after failing to receive the expected B-Ack frame.

The recipient may implement a timeout that enables it to stop waiting for missing old frames if appropriate, hence allowing new MSDUs to be released to the MAC client and some B-Ack buffer resources to be freed.

7.1.7 Duplicate detection

A received frame shall be treated by the recipient as a duplicate of the last frame received if the Retry bit in its MAC header is set to 1 and the two frames have identical values in the following fields of their MAC header: HID, NID, Protocol Version, Security Protection, Frame Type, Frame Subtype, Sequence Number, and Fragment Number.

7.1.8 Fragmentation and reassembly

A sender may fragment only MSDUs to be transferred in data type frames with the Ack Policy field set to N-Ack or I-Ack..

The sender shall not fragment any MSDU to more than mMaxFragmentCount fragments. All fragments, except the last one, of the same MSDU shall have the same length.

The sender shall not alter the length of the fragments of an MSDU, by refragmentation or recombination, in retransmitting them.

The sender shall not transmit any other fragments of an MSDU after discarding one fragment of the MSDU.

The sender shall set to 0 the Fragment Number field in the data type frame containing the first fragment of an MSDU. It shall set the field in the data type frame containing each subsequent fragment of the same MSDU to one plus the Fragment Number value in the data type frame containing the previous fragment. The sender shall not alter the Fragment Number field in a data type frame in retransmitting the frame.

The sender shall set the Sequence Number field in all data type frames containing the fragments of the same MSDU to the same value.

A recipient shall completely reassemble an MSDU in the correct order before delivery to the MAC client. The recipient shall discard any MSDU with missing fragments.

7.2 Beacon frame transmission

A hub shall transmit a beacon frame, also referred to as a beacon in this standard, in every beacon period as shown in Figure 35. The hub shall not change its beacon period so long as one or more nodes are connected with it.

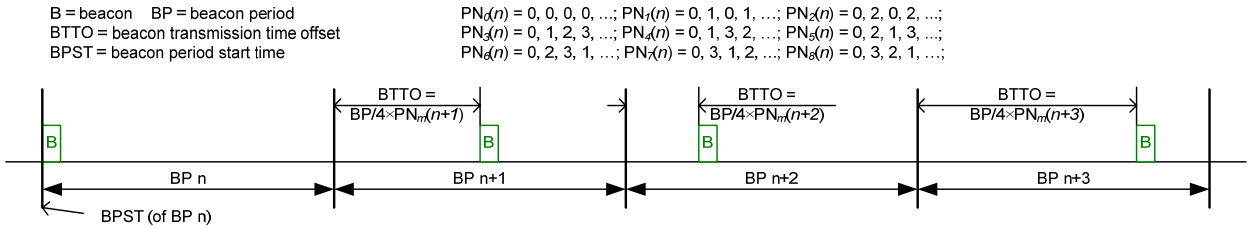


Figure 35 — Beacon transmission

The hub shall start transmitting a beacon out of its physical layer (PHY) at a time $t = PN_m(n) \times BP/4$ relative to the start of beacon period n. Here, m is the beacon shifting sequence index that the hub has chosen for its beacon transmission time pattern across beacon periods, BP is the length of its beacon period, and n is the phase of the chosen sequence ($n = 0, 1, \ldots$). The beacon shifting sequence index and phase values are contained in the beacon as defined in 6.3.1.2.

The values of the beacon shifting sequence of index 0 are identical to 0, i.e., $PN_0(n) = 0$, $n = 0, 1, \ldots$. With this sequence, the beacon transmission time always occurs at the start of each beacon period. A hub should choose a beacon shifting sequence that is not being used by its neighbor hubs to mitigate potential repeated beacon collisions and scheduled allocation conflicts between overlapping or adjacent subnets operating on the same channel.

7.3 Unconnected exchange

The hub should provide unconnected polled allocations to unconnected nodes, some of which might not be capable of using CSMA/CA based random access, for the transmission of their first frame to it. An unconnected polled allocation is a polled allocation provided by the hub through a poll that is addressed to Unconnected_Broadcast_NID, i.e., the NID field of the MAC header of the poll frame is set to Unconnected_Broadcast_NID.

An unconnected node may send its first frame, but not subsequent frames, to the hub in the next unconnected polled allocation. If the node does not receive an acknowledgment, it may retry the first frame in the next polled allocation with a probability $P = \max(1/4, 1 - R/4)$, where R is the number of the retry, i.e., R equals 1 for the first retry, 2 for the second retry, and so on.

An unconnected node may send to a hub its first frame, or the retry thereof, before knowing the full IEEE MAC address of the hub or before receiving a beacon from the hub. In this case, it shall set the Recipient Address field in the frame according to the exception stated in 7.1.2.

Once the hub receives and acknowledges the first frame sent from an unconnected node and assigns a unique Unconnected_NID to the node, it should provide polled allocations specifically addressed to this Unconnected_NID, as described in 7.5.1, until the hub has received a Connection Request frame from the node.

Before the node receives a Connection Assignment frame from the hub, it should stay in an active state to be ready to receive or transmit a frame.

A node may also use CSMA/CA based random access as described in 7.6, if capable, to transmit its frames before and after its connection with a hub.

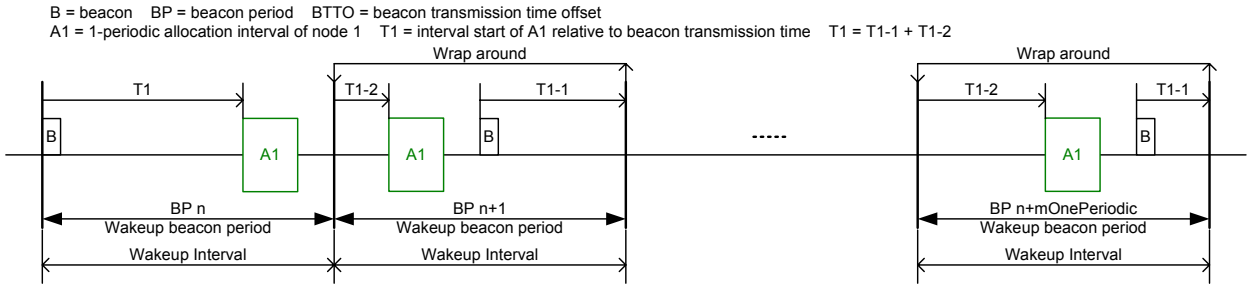
7.4 Scheduled access

A node and a hub shall employ scheduled access as described below, if they need to obtain 1-periodic or m-periodic allocations for periodic contention-free frame exchanges within their subnet. A node shall not have both 1-periodic and m-periodic allocations at the same time.

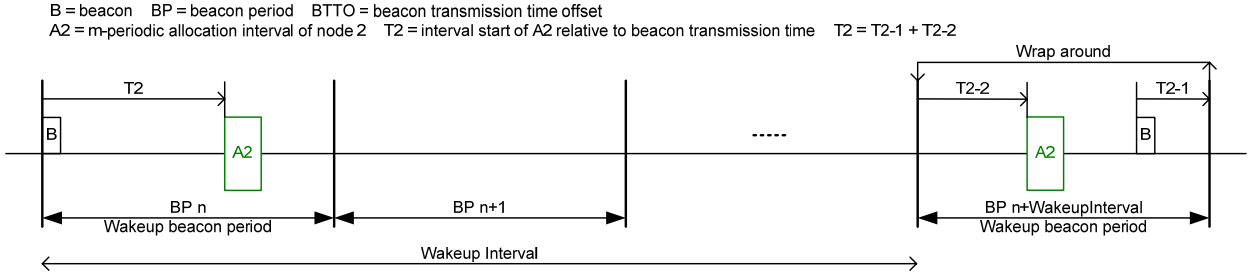
To have a 1-periodic allocation, which has one or more allocation intervals spanning the same allocation slots in every beacon period, a node shall wake up in every beacon period, i.e., the node shall treat every beacon period as its wakeup beacon period, as illustrated in Figure 36(a).

To have an m-periodic allocation, which has one or more allocation intervals spanning the same allocation slots in every m > 1 beacon periods, a node shall wakeup in the beacon periods containing those allocation intervals, i.e., the node shall treat one beacon period out of every m > 1 beacon periods as its wakeup beacon period, as illustrated in Figure 36(b).

Note that the location of a given allocation slot, and hence a given allocation interval, shifts with the beacon start time across beacon periods as defined in Figure 14. This temporal shift is based on the beacon transmission time pattern selected by the hub.



(a) Example 1-periodic allocation



(b) Example m-periodic allocation

Figure 36 — Scheduled allocations

7.4.1 Starting scheduled allocations

To obtain one or more new scheduled allocations, a node shall send a Connection Request frame to the hub when permitted to do so, setting the Wakeup Interval field in the frame to 1 for 1-periodic allocations and to m > 1 for m-periodic allocations, and including an Uplink Request IE or/and a downlink Request IE in the frame for uplink or/and downlink allocations, respectively, with the Minimum Interval Length and Minimum Allocation Length fields set to nonzero values.

To grant scheduled allocations, the hub shall respond with a Connection Assignment frame to the node, including an Uplink Assignment IE or/and a downlink Assignment IE in the frame for uplink or/and downlink allocations, respectively, in the frame with different values assigned to the Interval Start and Interval End fields.

7.4.2 Using scheduled allocations

Upon receiving the Connection Assignment frame as described in 7.4.1, the node may start initiating frame transactions in the corresponding uplink allocation intervals granted to it in its next wakeup beacon period, as illustrated in Figure 37. The hub shall be ready to receive the frames transmitted by the node and acknowledge them when appropriate within the allocation intervals.

Likewise, upon successfully sending the Connection Assignment frame, the hub may start initiating frame transactions in the corresponding downlink allocation intervals granted to the node in the node's next wakeup beacon period, as also illustrated in Figure 37. The node shall be ready to receive the frames transmitted by the hub and acknowledge them when appropriate within the allocation intervals.

The node or the hub shall transmit an acknowledgment frame, I-Ack or B-Ack, when required, TIFS after the end of the previous frame. The node or the hub may initiate another frame transaction in a scheduled uplink or downlink allocation interval, respectively, TIFS after the end of the expected acknowledgment frame regardless of whether it received the acknowledgment frame. The node and the hub shall ensure that the frame transactions, including acknowledgment frames if required, in their scheduled allocations stay within their scheduled uplink or downlink allocation intervals, respectively, taking the appropriate guardtime into account.

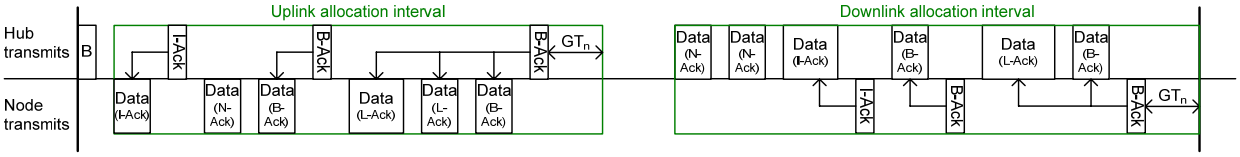


Figure 37 — Example frame transactions in scheduled uplink and downlink allocation intervals

7.4.3 Modifying scheduled allocations

A node may modify existing scheduled allocations by sending another Connection Request frame specifying the new requirements. The hub shall treat this request as a new request, except that it shall set the Change Indication field in its responding Connection Assignment frame with reference to the last Connection Assignment frame it sent to the node. In particular, if the hub rejects the modifications but maintains the existing allocations, it shall respond with a Connection Assignment frame with the Change Indication field set to 0 and the other fields kept to the respective values contained in the last Connection Assignment frame sent to the node.

A hub may, but should not where possible, modify scheduled allocations of a node on its own by sending the node an unsolicited Connection Assignment frame specifying the new assignments to those allocations, setting the Change Indication field in the frame with reference to the last Connection Assignment frame it sent to the same node.

7.4.4 Aborting scheduled allocations

A node or a hub shall treat an existing 1-periodic uplink or downlink allocation, respectively, to have been aborted after failing to receive any expected acknowledgment frame in the allocation in the last mOnePeriodic beacon periods. Likewise, a node or a hub shall treat an existing m-periodic uplink or downlink allocation, respectively, to have been aborted after failing to receive any expected acknowledgment frame in the allocation in the last mMultiPeriodic wakeup beacon periods. A node shall transmit or retransmit at least one management or data type frame with the Ack Policy field set to I-Ack or B-Ack in each of its wakeup beacon periods, so as to reduce the chance of experiencing an abortion of its scheduled allocations.

Further, a hub or a node shall treat an existing 1-periodic uplink or downlink allocation, respectively, to have been aborted after failing to receive any frame in the allocation from the expected sender in the last mOnePeriodic beacon periods. Likewise, a node or a hub shall treat an existing m-periodic uplink or downlink allocation, respectively, to have been aborted after failing to receive any frame in the allocation from the expected sender in the last mMultiPeriodic wakeup beacon periods.

Subsequently, the hub may reclaim the aborted scheduled allocations.

A node and a hub may start a new scheduled allocation procedure as specified in 7.4.1 to reinstate their lost allocations or obtain their replacements.

7.4.5 Ending scheduled allocations

A node may at any time end scheduled allocations by sending a modified Connection Request frame that contains Allocation Request fields with the Allocation ID fields identifying those allocations and with the Minimum Interval Length and Minimum Allocation Length fields set to 0.

A hub may, but should not where possible, at any time end any scheduled allocations of a node by sending the node a modified Connection Assignment frame that contains Allocation Assignment fields with the Allocation ID fields identifying those allocations and the Interval Start and Interval End fields set to 0.

A node or a hub may send a Disconnection frame to end all of their scheduled allocations and relinquish the node's Connected_NID.

7.5 Improvised access

A node and a hub shall employ improvised access through polls and posts as characterized in Table 13and described below, if they need to obtain polled and posted allocations for on-demand contention-free frame exchanges within their subnet. A polled or posted allocation contains a time interval that does not reoccur in subsequent beacon periods without the hub invoking another instance of improvised access.

Table 13 — Frames and fields sent by a hub enabling polls and posts

	Short distance	Long distance
Poll	Poll, I-Ack+Poll, or B-Ack+Poll frame, with More Data = 0, Sequence Number = *A*, Fragment Number = *Reserved* (see. 6.2.1.3.8, 6.2.1.3.10, and 6.2.1.3.11): A polled allocation starts TIFS after the end of the current frame, and ends at the end of allocation slot *A* located in the current beacon period.	Poll, I-Ack+Poll, or B-Ack+Poll frame, with More Data = 1, Sequence Number = *A*, Fragment Number = *B* (see. 6.2.1.3.8, 6.2.1.3.10, and 6.2.1.3.11): No polled allocation follows this frame, but another poll starts at the start of allocation slot *A* located in the current beacon period if *B* = 0 or in the next *B*th beacon period if *B* > 0.
Post	Non-beacon management or data type frame, with More Data = 1 (see. 6.2.1.3.8): A post starts TIFS after the end of the current frame transaction. A poll providing another poll but not a polled allocation may be considered a post.	I-Ack or B-Ack frame, with More Data = 1, Sequence Number = *A*, Fragment Number = *B* (see. 6.2.1.3.8, 6.2.1.3.10, and 6.2.1.3.11): A post starts at the start of allocation slot *A* located in the current beacon period if *B* = 0 or in the next *B*th beacon period if *B* > 0.

7.5.1 Polled allocations

A hub may send polls and grant polled allocations to a node only if both of them support polls as indicated in their respective MAC Capability fields (i.e., with the Poll Support field set to 1). A hub may send polls granting polled allocations to Unconnected_Broadcast_NIDs any time outside beacons and scheduled allocations, as described in 7.3 for unconnected exchange, without regard to the poll support setting in the MAC capability indicated by the nodes in its subnet.

7.5.1.1 Starting a polled allocation

To obtain a polled allocation for initiating another frame transaction, a node shall set the More Data field to 1 in the frame it is transmitting. The node should also set the Ack Policy field to I-Ack or B-Ack in some management or data type frames being transmitted. This enables the hub to send the node a short-distance or long-distance poll at an announced time through an I-Ack+Poll or B-Ack+Poll frame as described in Table 13.

To grant a polled allocation to a node, a hub shall send to the node a Poll frame when appropriate or an I-Ack+Poll or B-Ack+Poll frame when required to return an acknowledgment. The hub may send a Poll frame at the time indicated in a frame previously sent to the node.

Figure 38 illustrates instances of polls and polled allocations.

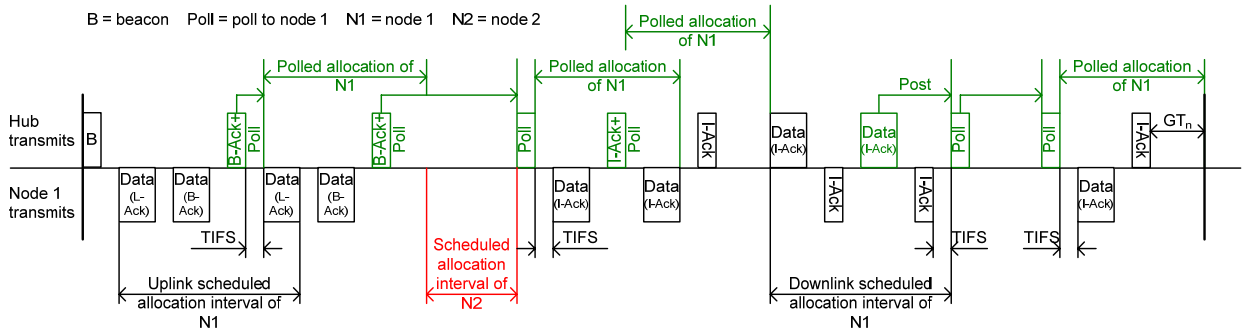


Figure 38 — Example polls and polled allocations

7.5.1.2 Using a polled allocation

A node that has indicated support for polls shall receive Poll frames at times that have been announced in frames previously sent to it by the hub, as described in Table 13. The node shall also receive I-Ack+Poll and B-Ack+Poll frames at times expected for acknowledgment arrivals. The node shall initiate a frame transaction at the start of a polled allocation granted to it through a preceding Poll, I-Ack+Poll, or B-Ack+Poll frame, as also illustrated in Figure 38.

A hub shall transmit an acknowledgment frame I-Ack or B-Ack when required, or I-Ack+Poll or B-Ack+Poll for another polled allocation, TIFS after the end of the previous frame in a polled allocation. The node may start another frame transaction, for retransmitting old frames or /and transmitting new frames, in the polled allocation TIFS after the end of the expected acknowledgment frame regardless of whether it received the acknowledgment frame. The node shall ensure that the frame transactions, including acknowledgment frames if required, in a polled allocation stay within the polled allocation interval, taking the appropriate guardtime into account.

7.5.1.3 Modifying a polled allocation

A hub may modify a polled allocation of a node by sending the node another poll through an I-Ack+Poll or B-Ack+Poll frame within the polled allocation that expands the allocation interval, effectively granting a new polled allocation in place of the existing one.

7.5.1.4 Aborting a polled allocation

A node may relinquish the remaining allocation interval of a polled allocation by indicating that it has no more data to transmit in the frame being transmitted and then by transmitting no more frames in the interval.

A hub shall not reclaim a polled allocation on the basis that it has not received any frame in the allocation, unless the node has explicitly relinquished the allocation as described above.

7.5.1.5 Ending a polled allocation

A node may at any time decline future polled allocations to it by sending a modified Connection Request frame that contains a MAC Capability indicating no poll support.

A hub may at any time decline future polled allocations to all the nodes connected with it by indicating no poll support in the MAC Capability of its subsequent beacons.

7.5.2 Posted allocations

A hub may send posts to a node while granting posted allocations outside the scheduled allocation of the node so long as it has informed the node of the pending posts through previously transmitted frames.

7.5.2.1 Starting a posted allocation

To obtain a posted allocation for receiving a post, a node that does not have a scheduled downlink allocation shall send at least one management or data type frame with the Ack Policy field set to I-Ack or B-Ack in each of its wakeup beacon periods. This enables the hub to send the node a long-distance post containing essential network management information or a critical user message at a specific time through an I-Ack or B-Ack frame as described in Table 13.

To send a short-distance post to a node, a hub shall set the More Data field to 1 in the non-beacon management or data type frame it is transmitting, as also described in Table 13. To send a long-distance post to a node, a hub shall send to the node an I-Ack or B-Ack frame when required to return an acknowledgment, as noted in the above.

Figure 39 illustrates instances of posts and posted allocations.

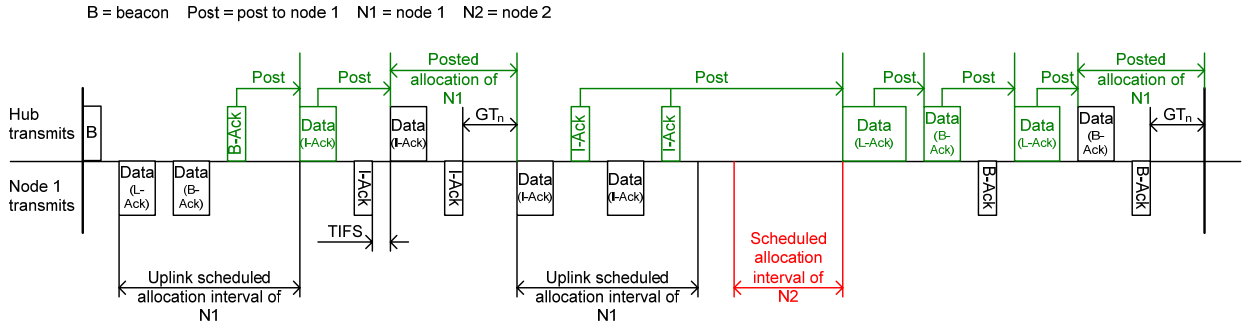


Figure 39 — Example posts and posted allocations

7.5.2.2 Using a posted allocation

A hub shall initiate frame transactions at the start of its posted allocations that have been announced in earlier frames as described in Table 13.

A node shall receive posts and participate in the frame transactions in its posted allocations that have been indicated to it earlier through other frames.

The node shall transmit an acknowledgment frame I-Ack or B-Ack when required, TIFS after the end of the previous frame in a posted allocation. If the hub has announced in the current frame that it has another short-distance post, it may initiate another frame transaction, for retransmitting old frames or/and transmitting new frames, TIFS after the end of the expected acknowledgment frame regardless of whether it received the acknowledgment frame. The hub shall ensure that the frame transaction, including the acknowledgment frame if required, in a posted allocation stays within the intended posted allocation interval, taking the appropriate guardtime into account.

7.5.2.3 Modifying a posted allocation

A hub shall not modify a posted allocation once it initializes a frame transaction in the allocation, as a posted allocation is just long enough for a frame transaction. A hub may grant itself another posted allocation for initializing another frame transaction after informing the target node of the time of the new allocation through the current frame .

7.5.2.4 Aborting a posted allocation

A hub shall not abort a posted allocation once it initializes a frame transaction in the allocation.

7.5.2.5 Ending a posted allocation

A node shall not end a pending posted allocation nor potential future posted allocations, as it needs to receive all the posts—which might carry essential network management information or critical user messages—from the hub with which it is connected.

A hub shall not end a pending posted allocation once it has announced it through a frame.

7.6 Random access

A node shall employ random access based on CSMA/CA by maintaining a backoff counter as described below, if it needs to obtain contended allocations in random access phases (RAPs) for initiating contention-based frame transactions. The node shall initialize the backoff counter to 0. A contended allocation contains a time interval that does not reoccur in subsequent beacon periods without the node invoking another instance of random access.

If a hub has indicated support for random access (i.e., with its Poll Support field set to 1), it shall announce in each beacon up to two RAPs for the next beacon period, as illustrated in Figure 40. One of them shall start immediately after the end of the beacon frame, and shall have a length RAP1 such that the combined duration of the beacon and RAP1 is not less than the minimum B+RAP1 length communicated earlier to the nodes connected with the hub through the hub's Connection Assignment frames. The other shall have a length RAP2 ≥ 0, and shall start half a beacon period away from the start of the beacon in the same beacon period if RAP2 > 0. The hub should ensure that the RAPs it announced do not overlap with any scheduled allocations.

If a hub has indicated no support for random access (i.e., with its Poll Support field set to 0), it shall announce no RAPs in any beacon period.

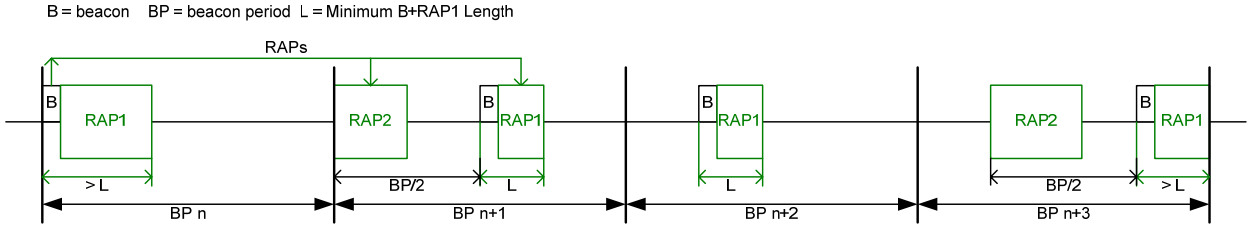


Figure 40 — Random access phases (RAPs)

The values for CWmin and CWmax used in obtaining a contended allocation as also described below shall be selected based on the user priority of the traffic to be transmitted in the allocation according to Table 14.

Table 14 — CWmin and CWmax values

Priority	User Priority	Traffic designation	CWmin	CWmax
Lowest	0		16	64
	1		16	32
	2		8	32
	3		8	16
	4		4	16
	5		4	8
	6		2	8
Highest	7	Emergency message	1	4

7.6.1 Starting a contended allocation

To obtain a new contended allocation, a node shall set the backoff counter to an integer sample of a random variable uniformly distributed over the interval [1, CW], when its backoff counter has a value of 0 and the node has at least one frame of user priority UP to transmit or retransmit, where CW is a contention window chosen as follows.

- If the node has not obtained any contended allocation previously, it shall set the CW to CWmin[UP].
- If the node has succeeded, i.e., if the node has received an expected acknowledgment, I-Ack or B-Ack frame, to its last frame transmission in the current contended allocation, it shall set the CW to CWmin[UP] as well.
- If the node is transmitting a frame requiring no acknowledgment, either an I-Ack or a B-Ack frame, in the current contended allocation, it shall keep the CW unchanged.
- If the node has failed, i.e., if the node has not received an expected acknowledgment at the expected time,
  - it shall keep the CW unchanged if this is the mth time the node failed since it last succeeded, where m is an odd number;
  - it shall double the CW if this the nth time the node failed since it last succeeded, where n is an even number.;
- If doubling the CW would exceed CWmax[UP], the node shall set the CW to CWmax[UP].

The node shall lock the backoff counter when any of the following events occurs:

- The backoff counter is reset upon decrementing to 0.
- The channel is busy. If the channel is busy because the node detected a frame transmission, the channel remains busy until at least the end of the frame transmission without the node having to re-sense the channel.
- The current time is outside any RAP.
- The current time is at the start of a contention slot inside a RAP but the time between the end of the slot and the end of the RAP is not long enough for completing a frame transaction and setting aside a nominal guardtime mGT_Nominal.

The node shall unlock the backoff counter when both of the following conditions are met:

- The channel has been idle for TIFS.
- The current time is at the start of a contention slot inside a RAP and the time between the end of the slot and the end of the RAP is long enough for completing a frame transaction plus a nominal guardtime mGT_Nominal.

The first contention slot before the node starts or resumes decrementing its backoff counter shall start once the channel has been idle for TIFS. A successive contention slot shall start once the current contention slot ends, until the node locks the backoff counter again, as illustrated in Figure 41. Each contention slot shall have a fixed duration of SlotLength.

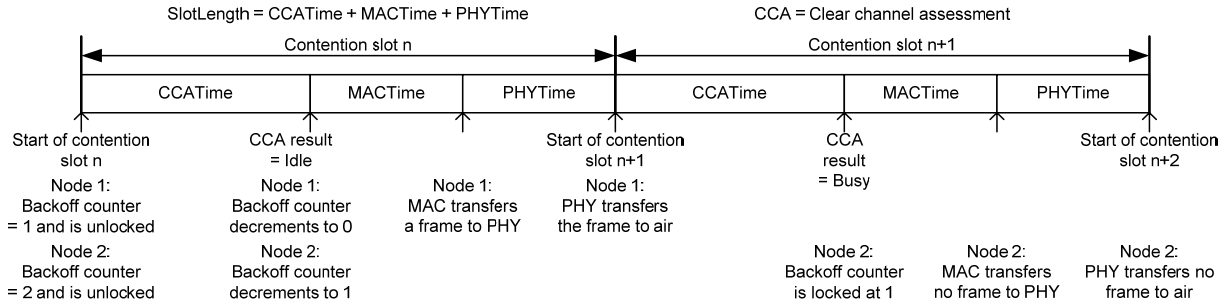


Figure 41 — Contention slot structure

The node shall decrement its backoff counter by one for each idle contention slot if the backoff counter is unlocked, as also shown in Figure 41. In particular, the node shall treat a contention slot to be idle if it determines that the channel has been idle between the start of the slot and CCATime later, decrementing the backoff counter effectively CCATime after the start of the slot, so that the node will transmit a frame to the transport medium (i.e., air) at the end of the slot in case its backoff counter reaches 0, as further described below.

If the backoff counter reaches 0 in the current contention slot, the node shall have obtained a contended allocation that starts at the end of the current slot with a maximum length of mCADurationLimit. Figure 42 illustrates how to start and use contended allocations.

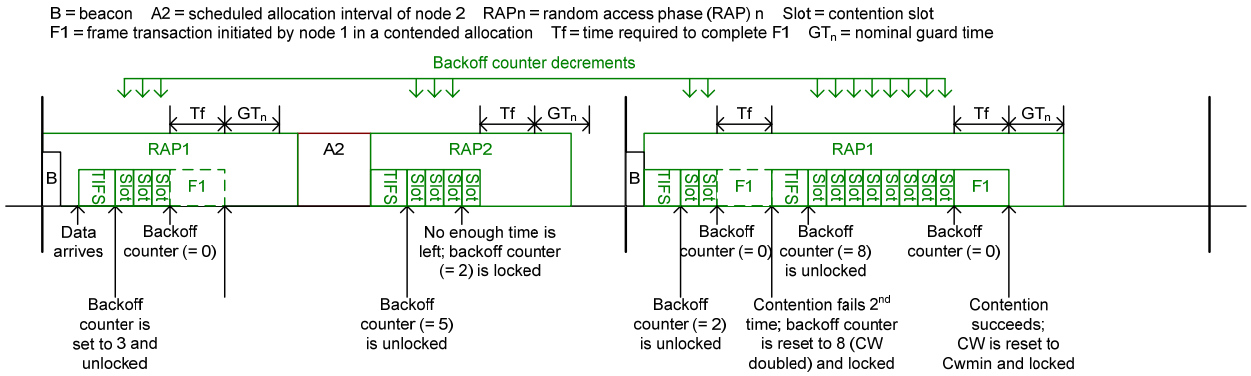


Figure 42 — Example random access

7.6.2 Using a contended allocation

A node shall transmit a frame for which it obtained the contended allocation at the start of the allocation. The recipient shall transmit an acknowledgment frame, I-Ack or B-Ack, when required, TIFS after the end of the previous frame.

To resume its transmission following a received acknowledgment frame, the node shall transmit a new frame or retransmit an old frame TIFS after the end of the acknowledgment frame. The node shall end its transmission in the allocation such that the last transmission in the allocation completes at least the nominal guardtime GT_n = mGT_Nominal earlier than the end of the allocation or of the current RAP, whichever is earlier.

7.6.3 Modifying a contended allocation

The node may not modify the contended allocation.

7.6.4 Aborting a contended allocation

A node shall treat a contended allocation to have been aborted after failing to receive the expected acknowledgment frame in the allocation.

A node may start a new contended allocation procedure as specified in 7.6.1 to obtain another contended allocation.

7.6.5 Ending a contended allocation

A node may at any time end a contended allocation by not following with another frame transmission in the allocation.

7.7 Clock synchronization and guardtime provisioning

A node or a hub shall maintain a clock with a minimum resolution of mClockResolution and with a minimum accuracy of mClockAccuracy to time its frame transmission and reception.

A node or a hub shall time its transmission and reception in any of its allocation intervals according to its local clock, setting aside appropriate GTs as specified below and illustrated in Figure 43:

– The hub shall commence its beacon transmission at the nominal start of the beacon transmission.

– The hub shall commence its transmission in a downlink allocation interval at the nominal start of the interval, and the hub shall end its transmission in the interval early enough such that the last transmission in the interval completes at least a guardtime of GT_n prior to the nominal end of the interval.

– The hub shall commence its reception in an uplink allocation interval at least GT_n prior to the nominal start of the interval.

– If a node's last synchronization to the hub was less than SI_n ago at the nominal start of its next uplink allocation interval, the node shall commence its transmission in the interval at that nominal start time, and the node shall end its transmission in the interval early enough such that the last transmission in the interval completes at least a guardtime of GT_n prior to the nominal end of the interval.

– If a node's last synchronization to the hub was less than SI_n ago at the nominal start of the next beacon transmission, the node shall commence its reception of the beacon at least GT_n prior to that nominal start time.

– If a node's last synchronization to the hub was less than SI_n ago at the nominal start of its next downlink allocation interval, the node shall commence its reception in the interval at least GT_n prior to that nominal start time.

– If a node's last synchronization to the hub was $SI_n + SI_a$ ago at the nominal start of its next scheduled uplink allocation interval, the node shall commence its transmission in the interval a guardtime of GT_a later than that nominal start time, and the node shall end its transmission in the interval early enough such that the last transmission in the interval completes at least a guardtime of $GT_n + GT_a$ prior to the nominal end of the interval.

– If a node's last synchronization to the hub was $SI_n + SI_a$ ago at the nominal start of its next scheduled downlink allocation interval, the node shall commence its reception in the interval at least $GT_n + GT_a$ earlier than that nominal start time.

A node shall synchronize to its hub through the beacons or the first frames in scheduled allocation intervals received from the hub. In particular, the node shall advance or delay its clock by a total amount of

$$D = T_S - T_L, \text{ if } T_S > T_L$$

or

$$D = T_L - T_S, \text{ if } T_S < T_L$$

respectively, where T_S is the time when a beacon frame or a frame with the "First Frame" bit in its MAC header set to 1 was scheduled to start on the transport medium (i.e., air), and T_L is the time when the frame was received according to the local clock.

The various guardtime components shall be determined as follows:

$$\text{mGT\_Nominal} = GT_n = GT_0 + 2\times D_n, \quad D_n = SI_n \times \text{mClockAccuracy}$$
$$GT_0 = \text{TIFS} + \text{mSynchResolution} + \text{mTxResolution}, \quad \text{mTxResolution} = \text{mClockResolution}$$
$$GT_a = 2\times D_a, \quad D_a = SI_a \times \text{mClockAccuracy}$$

The parameter GT_n = mGT_Nominal has a predefined value as listed in Table 15, and designates the guardtime nominally observed by a node or a hub. The parameter GT_0 comprises the RX-TX or TX-RX turnaround time TIFS, the synchronization error mSynchResolution, and the transmission timing error mTxResolution, which are all of fixed values that are independent of clock drifts and listed in Table 15 as well. Thus the value of GT_0 is also predefined. Given the values of GT_n and GT_0, the parameter D_n represents the nominally allowed clock drift of a node or hub relative to an ideal (nominal) clock. With the minimum clock accuracy of mClockAccuracy also given in Table 15, D_n is predetermined too; it delimits a nominal synchronization interval SI_n over which the clock drift of any node is accounted for in the nominal guardtime GT_n.

The parameter SI_a denotes the synchronization interval additional to SI_n for a synchronization. The corresponding additional clock drift D_a is a function of SI_n and accounts for the required additional guardtime GT_a. The values of these two parameters are specific to the node and time of concern.

An illustration of clock drift and guardtime provisioning is given in Figure 43.

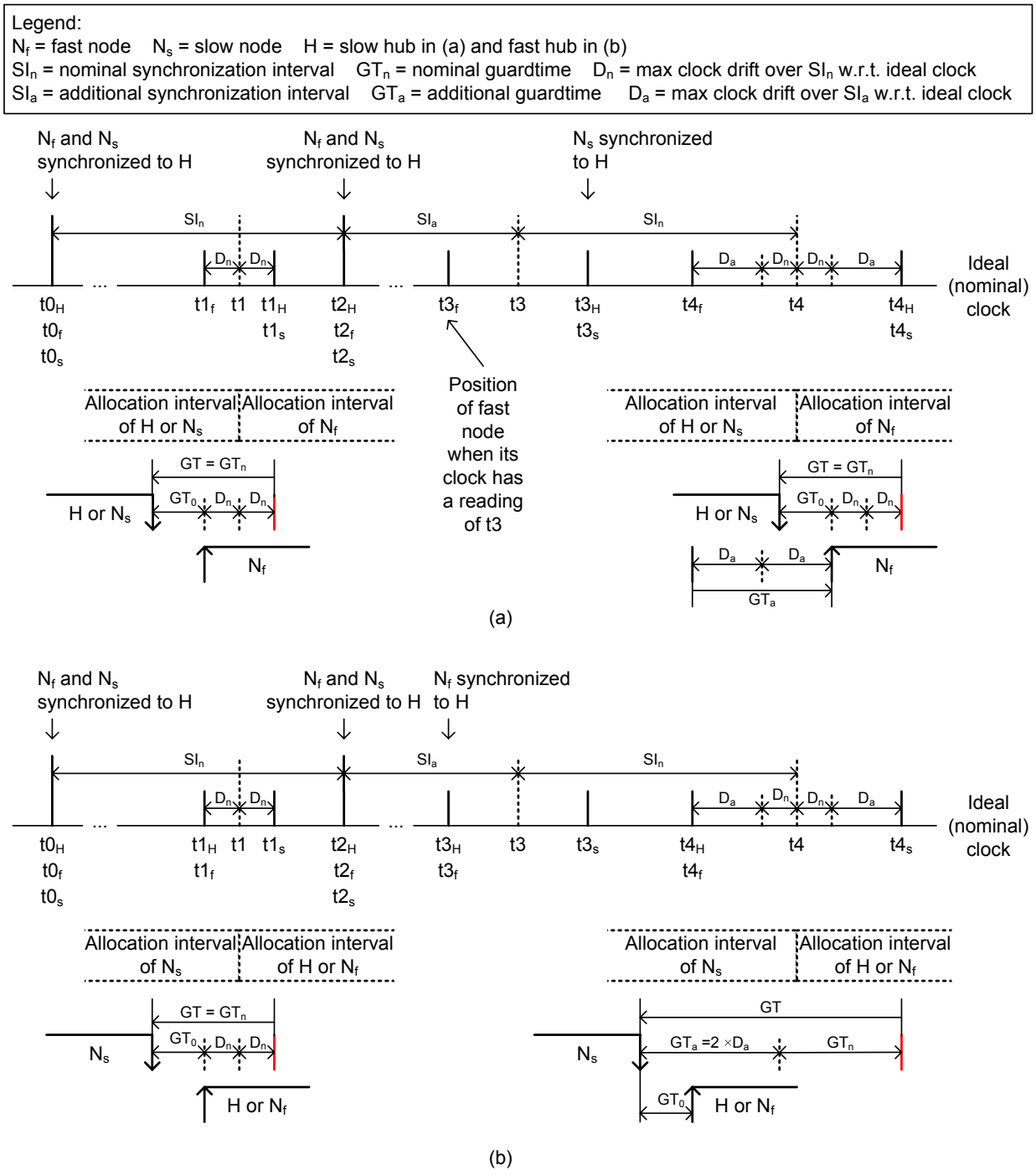


Figure 43 — Clock drifts and guartimes

7.8 Power management

A node may hibernate across a number of beacon periods, and may sleep over some time intervals even in its wakeup beacon periods, as clarified below.

7.8.1 Hibernation—macroscopic power management

To hibernate—without receiving or transmitting any traffic—in some beacon periods, the node shall set the Wakeup Interval field in its last Connection Request frame to an integer larger than 1, while setting the Next Wakeup field in the frame to a value specifying its intended next wakeup beacon period.

To wakeup—for receiving or/and transmitting frames—in every beacon period, the node shall set the Wakeup Interval field in its last Connection Request frame to 1, while setting the Next Wakeup field in the frame to a value identifying the next beacon period.

The intended recipient hub of the Connection Request frame should honor the values of the received Wakeup Interval and Next Wakeup fields whenever possible, but may set them to different values if need be, in its responding Connection Assignment frame. The hub may later modify these values by sending to the node another Connection Assignment frame if warranted by new conditions.

If the hub sets the Wakeup Interval field in its responding frame to an integer larger than 1, it may grant only m-periodic allocations to the node, with the allocation intervals being in the node's wakeup beacon periods, in accordance with the node's last connection request whenever possible, but shall not grant to the node any 1-periodic allocations.

If the hub sets the Wakeup Interval field in its responding frame to 1, it may grant only 1-periodic allocations to the node, with the allocation intervals being in every beacon period, in accordance with the node's last connection request whenever possible, but shall not grant to the node any m-periodic allocations.

If the Wakeup Interval value in the Connection Assignment frame last received from the hub is larger than 1, the node shall wake up in each of its wakeup beacon periods based on the latest Wakeup Interval and Next Wakeup values provided in that frame by the hub, to transmit or/and receive frames in the granted m-periodic allocation intervals, and to receive the beacon if needed.

If the Wakeup Interval value in the Connection Assignment frame last received from the hub is 1, the node shall wake up in every beacon period, to transmit or/and receive frames in the granted 1-periodic allocation intervals, and to receive the beacon if needed.

Figure 44 illustrates macroscopic power management across beacon periods.

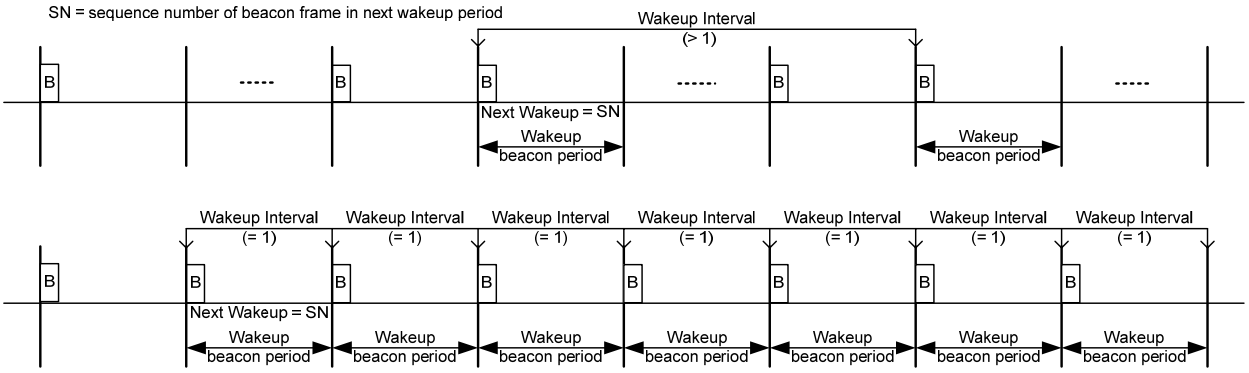


Figure 44 — Macroscopic power management

7.8.2 Sleep—microscopic power management

A node shall wake up to receive a beacon from a hub when it needs a beacon reception to synchronize with the hub or to obtain certain information contained in a beacon.

A node shall wake up to receive and transmit frames in its scheduled allocations in its wakeup beacon periods, as illustrated in Figure 45.

In addition, the node shall stay active participating in frame transactions in its expected posted allocations, as illustrated in Figure 45(a). The hub should arrange to have the posted allocations of a node to occur in the node's wakeup beacon periods, if possible. If the node did not receive a frame at the announced time for a pending post, it should stay in receive mode until the hub could have finished a frame transaction for the post and retransmitted a frame TIFS later, as also illustrated in Figure 45(a).

If the node has indicated its support for polls through its MAC Capability field of its last Connection Request frame, it shall also stay active in such times as to receive announced polls and initiate frame transactions in its polled allocations, as illustrated in Figure 45(b). The hub should arrange to have the polled allocations of a node to occur in the node's wakeup beacon periods, if possible.

Outside the time intervals noted in the above, the node may sleep—without receiving or transmitting any traffic.

Figure 45 illustrates microscopic power management in a wakeup beacon period .

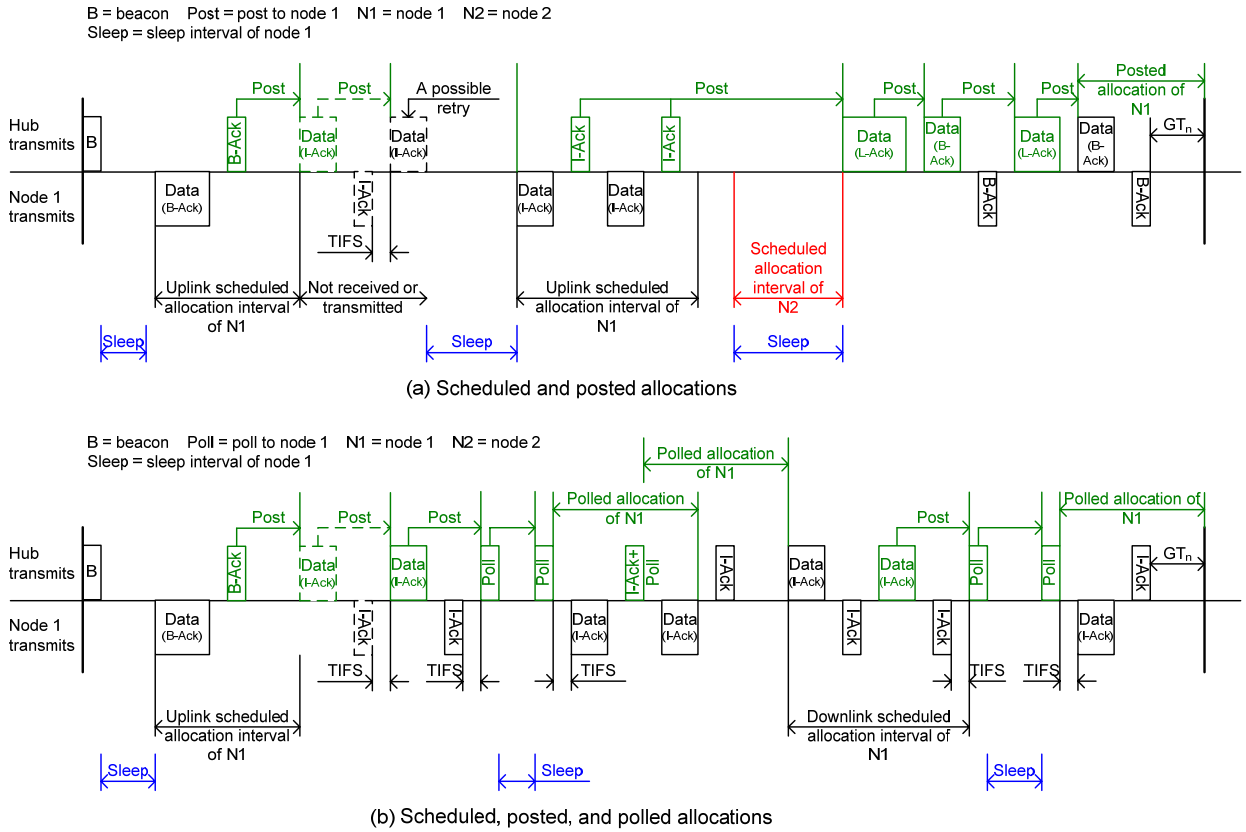


Figure 45 — Microscopic power management

7.9 Channel hopping

Unless for regulatory compliance, a hub shall hop to the next channel after dwelling on the current channel for a fixed number of beacon periods as communicated to the nodes connected with the hub through Connection Assignment frames. The hub shall not hop to a new channel in the middle of a beacon period.

A hub shall generate a channel hopping sequence based on the maximum-length Galois linear feedback shift register (LFSR) defined in Figure 46 and by the following generator polynomial

$$x^{16} + x^{14} + x^{13} + x^{11} + 1$$

The state of the LFSR at stage k is given by

$$Y_k = 2^0 \times r_{k,0} + 2^1 \times r_{k,1} + \ldots + 2^{15} \times r_{k,15}$$

Y_k represents the binary value read from the bits $r_{k,0}$, $r_{k,1}$, …, $r_{k,15}$ of the individual registers at stage k, with $r_{k,0}$ being the least-significant bit and $r_{k,15}$ being the most-significant bit.

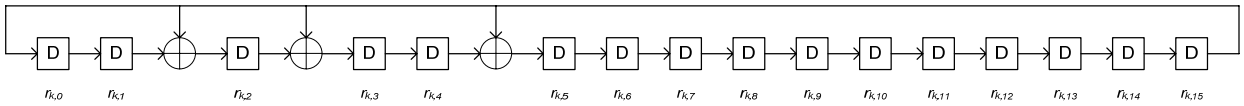


Figure 46 — 16-bit Galois LFSR for channel hopping sequence generation

Given the current state Y_k of the LFSR, the hub shall generate the next state Y_{k+1} of the LFSR, i.e, the state of the LFSR at the next stage, stage $k+1$. Accordingly, the hub shall generate the channel number C_{k+1} of the next channel from the channel number C_k of the current channel and the next state Y_{k+1} of the LFSR as follows:

$$Z_{k+1} = Y_{k+1} \bmod N_{reduced}$$

$$C_{k+1} = (C_k + Z_{k+1} + N_{sep}) \bmod N_{ch}$$

In the above, the notation mod denotes modular operation. N_{ch} = pChannelsTotal is the number of total channels in the operating frequency band as listed in Table 16. N_{sep} = pChannelSeparation is the minimum number of channels separated between two consecutive hops as illustrated in Figure 47. $N_{reduced}$ is the number of channels available for each hop on account of the channel separation constraint N_{sep}, and is given by

$$N_{reduced} = N_{ch} - 2N_{sep} + 1$$

The channels are numbered from 0 through $N_{ch} - 1$. Given the current channel number C_k, the next channel number C_{k+1} will be such that $|C_{k+1} - C_k| \geq N_{sep}$. All channels are selected with equal probability, as also shown in Figure 47.

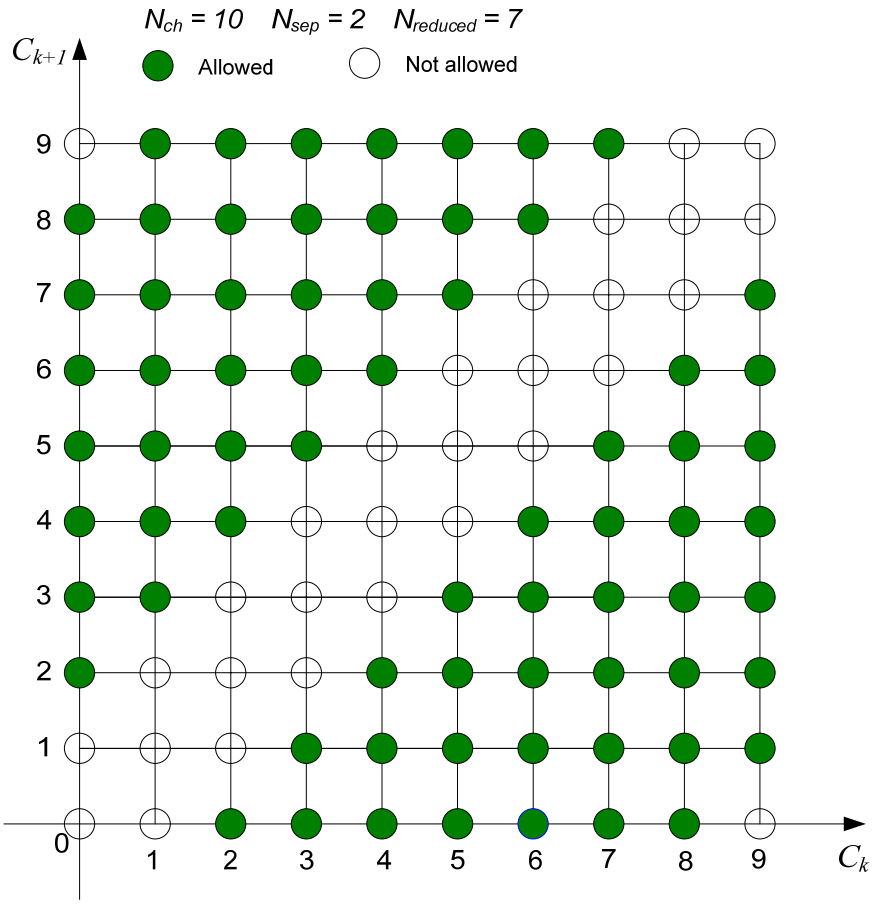


Figure 47 — Channel separation in consecutive hops

The hub shall set the initial state Y_0 of its LFSR to the 16 least-significant bits of its IEEE MAC address, with $r_{0,0}$ corresponding to the least-significant bit. The hub shall select the channel number C_0 of its initial channel as follows:

$$C_0 = Y_0 \bmod N_{ch}$$

To remain connected with a hub, a node shall hop to the same channel as the hub.

If required for regulatory compliance, a hub shall select a channel based on the applicable regulatory requirements, and shall dwell on the channel for an indefinite period of time. To communicate this selection to nodes, the hub shall set the channel hopping state to 0. To communicate with the hub, a node shall also dwell on the current channel for an indefinite period of time as well. This is equivalent to setting N_{sep} = pChannelSeparation to 0 in calculating the next channel number from the current channel number based on the equations given in the above.

7.10 Multi-rate support

A hub shall transmit beacons and polls that are addressed to Unconnected_Broadcast_NID at pBeaconDataRate.

A sender shall transmit frames at data rates supported by the intended recipient(s), based on information indicated in the recipient's PHY Capability IE.

7.11 Application Specific IE usage

A hub may include one or more Application Specific IEs at the end of its beacon.

A node may include one or more Application Specific IEs at the end of its Connection Request frame. A hub may also include one or more Application Specific IEs at the end of its Connection Assignment frame.

A recipient shall ignore unrecognized Application Specific IEs.

7.12 MAC sublayer parameters

Table 15 provides the values for the MAC sublayer parameters.

Table 15 — MAC sublayer parameters

mCADurationLimit	2 × Allocation Slot Length
mClockAccuracy	20 ppm
mClockResolution	10 μs
mGT_Nominal	Allocation Slot Length / 10
mMaxChannelChangeTime	256 beacon periods
mMaxFragmentCount	8
mMaxFrameBodyLength	pMaxFrameBodyLength
mMaxNumberSubnet	64
mMultiPeridoic	32
mOnePeridoic	32
mSynchResolution	pSynchResolution

Table 16 provides the values of the PHY dependent parameters used by the MAC sublayer.

Table 16 — PHY-dependent MAC sublayer parameters

pBeaconDataRate	64.5-129 kbps (band dependent)
pCCATime	63 × Symbol Period
pChannelSeparation	0 for no channel hopping
	TBD (band dependent)
pChannelsTotal	TBD (band dependent)
pClockAccuracy	20 ppm
pMaxFrameBodyLength	255 octets
pSlotLength	pCCATime + pTIFS
pSynchResolution	10 μs
pTIFS = TIFS	20 μs

8 Security services

This clause describes the security services used for secured frame exchanges at the MAC sublayer. Subclause 8.1 provides the mechanisms and rules for the selection and use of an appropriate security level. Subclause 8.2 specifies the facilities for the authentication and encryption of MAC frames to protect message authenticity, integrity, confidentiality, and privacy. Subclause 8.2.5 offers the measures for frame replay detection and filtering.

8.1 Security consideration

A hub shall transmit a beacon as an unsecured frame or as a secured frame authenticated, but not encrypted, by a group temporal key (GTK) distributed to the nodes that are secured with it.

A node and a hub shall follow the access state diagram Figure 3(a) for secured communication to exchange frames with each other if the hub's lowest security level required is not level 0 – unsecured communication – as indicated in the Security Requirement field of its latest beacon.

A node and a hub may follow the access state diagram Figure 3(a) for secured communication to exchange frames with each other even if the hub's lowest security level required is level 0 – unsecured communication – as indicated in its latest beacon.

A node and a hub may follow the access state diagram Figure 3(b) for unsecured communication to exchange frames with each other if the hub's lowest security level required is level 0 – unsecured communication – as indicated in its latest beacon. Upon finding some information specific to the node, such as through the IEEE MAC address of the node contained in the Sender Address field of a Connection Request frame received from the node, the hub may change its lowest security level required of the node to a higher level through the Security Requirement field of its Connection Assignment frame addressed back to the node. The node and the hub shall subsequently follow the access state diagram Figure 3(a) for secured communication to exchange frames with each other.

For secured communication, the node may indicate through the Security Requirement field of its Connection Request frame addressed to the hub a required lowest security level higher than the lowest security level required by the hub as indicated in the hub's latest beacon, as deemed necessary based on the information specific to the hub, such as through the IEEE MAC address of the hub contained in the Sender Address field of a beacon received from the hub. Likewise, the hub may indicate through the Security Requirement field of its Connection Assignment frame addressed back to the node a required lowest security level higher than the lowest security level required by the node as indicated in the node's connection request, as deemed necessary based on the information specific to the node.

For secured communication, the node may indicate through the Security Requirement field of its Connection Request frame addressed to the hub that control type frame authentication is required, even if this is not required by the hub as indicated in the hub's latest beacon. Likewise, the hub may indicate through the Security Requirement field of its Connection Assignment frame addressed back to the node that control type frame authentication is required, even if this is not required by the node as indicated in the node's connection request.

At the Secured or Connected State, the node and the hub shall exchange only secured frames with a security level that is the higher of the lowest security levels required by them as effective at the time of the frame transmission, with the following exceptions:

- GTK frames shall always be secured, both authenticated and encrypted.
- Poll frames shall never be authenticated or encrypted.
- Control type frames other than Poll frames
  - shall be neither authenticated nor encrypted, even when required to have security level 1 or 2, if both the hub and the node requires no control type frame authentication;
  - shall be authenticated but not encrypted, when required to have security level 1 or 2, if either the hub or the node requires control type frame authentication.

A node and a hub shall not further exchange non-control type frames with each other once the lowest security level required by one of them is found to be higher than the highest security level supported by the other as indicated in

the Security Support field of the Security Capability field of the latter's beacon, connection request, or connection assignment as appropriate.

A recipient shall ignore a received frame with an unexpected security level, other than responding with a control type frame if needed. A recipient shall also ignore a received secured frame with an invalid MIC, i.e., the MIC value calculated from the received frame as described in 8.2.5 is not the same as the MIC field contained in the received frame, except again for responding with a needed control type frame.

A recipient shall not treat a control-type frame received that was required to be authenticated but was not.

8.2 Frame authentication, encryption, and decryption

Secured frames shall be authenticated, and encrypted/decrypted when required, based on the CCM mode as specified in the NIST Special Publication 800-38C, with the AES forward cipher function for 128-bit keys as specified in FIPS Pub 197 applied as the underlying block cipher algorithm. The AES key used for a secured frame sent from a sender to a recipient shall be the pairwise temporal key (PTK) already established and currently used between the two parties, and the AES key used for a secured frame broadcast or multicast by a sender to a group shall be the group temporal key (GTK) already distributed and currently used by the sender for the broadcast or multicast group. A temporal key, PTK or GTK, shall be retired no later than when the Security Sequence Number field of the last frame secured by the key has reached the maximum value supported by the field.

The length of what is referred to as the Message Authentication Code (MAC) for message (frame) authentication in NIST Special Publication 800-38C but as the Message Integrity Code (MIC) in this standard—to be distinguished from another accustomed standing of the term MAC for medium access control—shall be four octets. That is, in the NIST Special Publication 800-38C, $t = 4$. Also, $q = 2$ shall be chosen as the octet length of the binary representation of the octet length of the frame payload.

The bit order of each input block to the CCM invocation and AES encryption shall be formatted as illustrated in Figure 48. It is the concatenation of the bits orders of the ordered octets of the constituent fields of the block, where the octet order of each constituent field is defined below, and the bits of each octet are ordered such that the most-significant bit (msb) is the first bit of the octet while the least-significant bit (lsb) is the last bit of the octet. The first octet or the first bit of a given component is shown on the left, and the last octet or the last bit is shown on the right, in the context of the component. The bit notations $input_0, \ldots, input_{127}$ correspond to those used for AES input block formation specified in FIPS Pub 197.

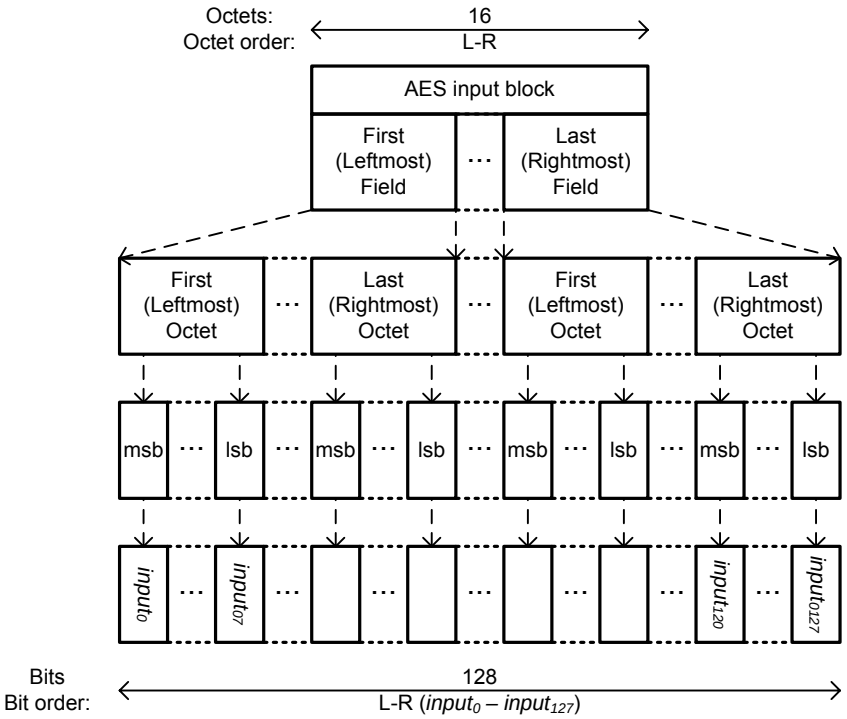


Figure 48 — Bit order for AES input blocks

8.2.1 Nonce formation

The Nonce as a required input field to each instance of CCM frame authentication and encryption/decryption is a 13-octet field that is formatted as shown in Figure 49. Here, the octets of the MAC header and Security Number fields are each ordered from left to right in accordance with their transmit order as defined in 6.1 and 6.2.

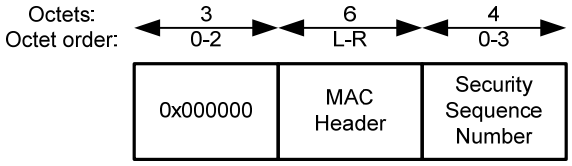
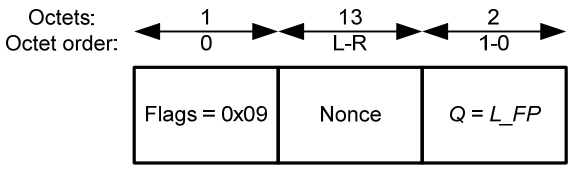


Figure 49 — Nonce format

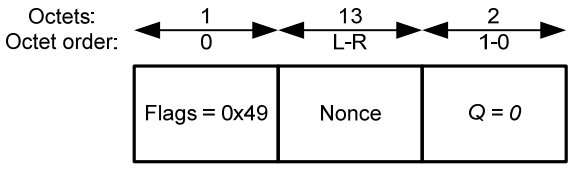
8.2.2 Initial block B_0 construction

The block B_0 as the first input block to the cipher block chaining (CBC) for frame authentication, i.e., MIC computation, is a 16-octet field that is formatted as shown in Figure 50. Here, $Q = L\_FP$ is the octet length of the frame payload as defined in Figure 10 and is encoded with the octet containing the most-significant bits on the left and the octet containing the least-significant bits on the right.

Only this block is present if the current frame does not have a frame payload.



(a) Frame payload encrypted



(b) Frame payload not encrypted

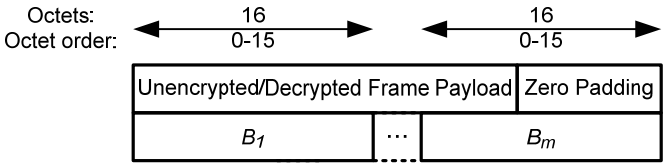
Figure 50 — Initial block B_0 format

8.2.3 Payload blocks B_1, …, B_m construction

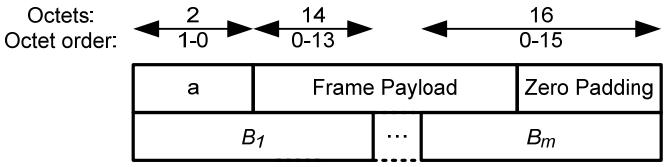
The blocks B_1, …, B_m as the subsequent input blocks to the CBC frame authentication, and also as the input blocks to the counter mode encryption/decryption, i.e., ciphertext computation and plain text recovery, if the frame payload is to be encrypted/decrypted, are each a 16-octet field that is formatted as shown in Figure 51. Here, $a = L\_FP$ is the octet length of the frame payload as defined in Figure 10 and is encoded with the octet containing the most-significant bits on the left and the octet containing the least-significant bits on the right, and the Frame Payload field is ordered from left to right in consistency with it transmit order as defined in 6.1 and 6.2.

These blocks are constructed from the unencrypted or decrypted frame payload. The last block contains one or more padded zero octets on the right end if the frame payload is not an integral multiple of 16 octets.

None of these blocks is present if the current frame does not have a frame payload.



(a) Frame payload encrypted



(b) Frame payload not encrypted

Figure 51 — Payload blocks B_1, …, B_m format

8.2.4 Counter blocks Ctr_0, …, Ctr_m formation

The block Ctr_0 as the input block to the counter mode encryption of the CBC output for MIC computation, and each of the blocks Ctr_1…, Ctr_m as the input blocks to the counter mode encryption/decryption if the frame payload is to be encrypted/decrypted, is a 16-octet field that is formatted as shown in Figure 52. Here, $i = 0$, …, m, respectively,

and is encoded with the octet containing the most-significant bits on the left and the octet containing the least-significant bits on the right.

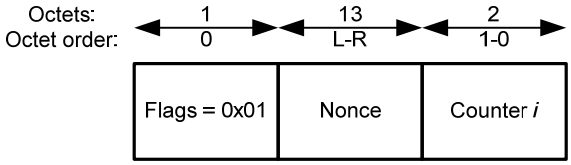


Figure 52 — Counter blocks Ctr_0, …, Ctr_m format

8.2.5 MIC construction

The MIC field in an authenticated frame is calculated as shown in Figure 53, where

$$MIC = LMB\_n(M), \quad M = AES(Ctr_0) \oplus X_m$$

$$X_0 = AES(B_0), \; X_i = AES(B_i \oplus X_{i-1}), \; i = 1, \ldots, m$$

Here, *LMB_n(M)* designates the *n* leftmost bits of the bit string *M*, the symbol ⊕ denotes bitwise exclusive-OR, and *AES*(*B*) represents the output of the forward cipher function of the AES block cipher algorithm applied to block *B* under the AES key PTK or GTK used to secure the frame. The MIC is ordered for transmission from its first octet on the left to its last octet on the right, as also illustrated in Figure 53. The octet notations out_0, …, out_{15} correspond to those used for AES output block formation specified in FIPS Pub 197.

The blocks required for the MIC computation are constructed from the unencrypted version of the frame to be transmitted at the sender side, and from the decrypted version of the received frame at the recipient side if the frame is encrypted.

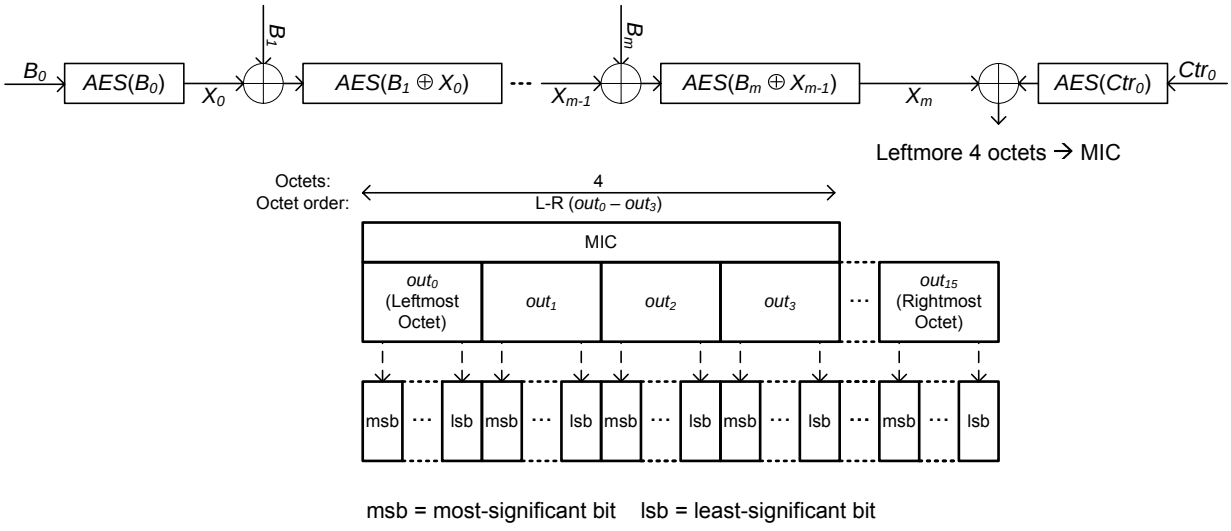


Figure 53 — MIC calculation and transmit order

8.2.6 Frame payload encryption

The encrypted frame payload in an encrypted frame is formatted as shown in Figure 54, where

$$B'_i = B_i \oplus AES(Ctr_i), \; i = 1, \ldots, m\text{-}1$$

$$B'_m = L\_n(B_m) \oplus L\_n(AES(Ctr_m))$$

Here, the symbol ⊕ denotes bitwise exclusive-OR, and *L_n(B)* designates the *n* leftmost octets of *B*. Moreover, AES(Ctr_i) represents the output of the forward cipher function of the AES block cipher algorithm applied to the counter block Ctr_i under the AES key PTK or GTK used to secure the frame. The encrypted frame payload has the same length as the unencrypted frame payload, so that $n \leq 16$ is the number of octets in B_m excluding the zero padding octets if any.

Each encrypted block is ordered for transmission from its first octet on the left to its last octet on the right, as also illustrated in Figure 54. The octet notations out_0, …, out_{15} correspond to those used for AES output block formation specified in FIPS Pub 197.

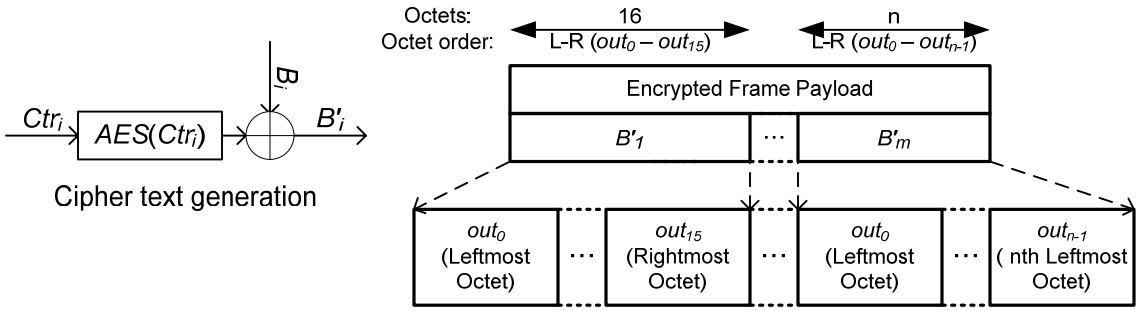


Figure 54 — Encrypted Frame Payload format for encrypted frames

8.2.7 Frame payload decryption

The frame payload in an encrypted frame is decrypted as shown in Figure 55, where

$$B_i = B'_i \oplus AES(Ctr_i),\ i = 1, \ldots, m\text{-}1$$

$$B^{-}_m = B'_m \oplus L\_n(AES(Ctr_m))$$

The decrypted frame payload has the same length as the encrypted frame payload, so that $n \leq 16$ is the number of octets in the last block B'_m of the encrypted frame payload received. The last decrypted block B^{-}_m is padded with $16 - n$ zero octets at the right end to form the last block B_m as shown in Figure 51(a) for MIC calculation over the received frame as described in 8.2.5.

Each decrypted block is ordered for MIC calculation, and delivery to the MAC client if the MIC is valid, from its first octet on the left to its last octet on the right, as also illustrated in Figure 55. Again the octet notations out_0, …, out_{15} correspond to those used for AES output block formation specified in FIPS Pub 197.

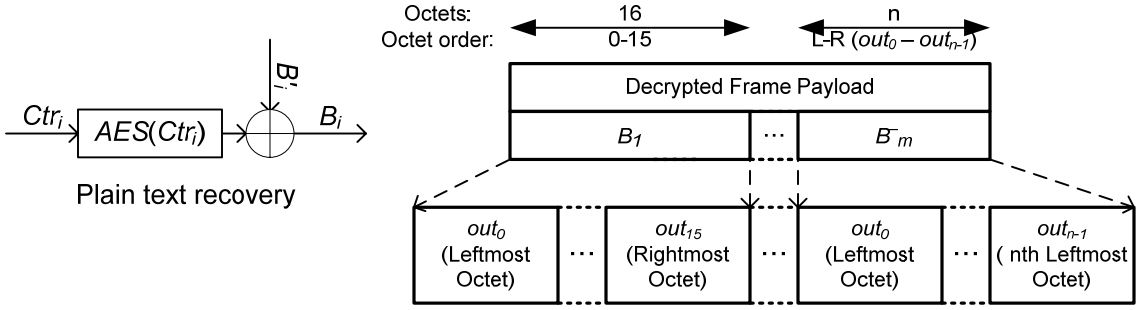


Figure 55 — Decrypted Frame Payload format for encrypted frames

8.3 Replay protection

A recipient shall treat a received frame with a valid MIC value, i.e., with the MIC field contained in the received frame equal to the MIC value calculated from the received frame, as a replay of a previously received frame if the Security Sequence Number field of the current frame has a value equal to or smaller than

– 0 if the current frame is secured by a pairwise temporal key (PTK) that has not been used for any previously received frames with a valid MIC field;

– GTK_SSN if the current frame is secured by a group temporal key (GTK) that has not been used for any previously received frames with a valid MIC field, where GTK_SSN is the value of the GTK SSN field contained in the Group Temporal Key (GTK) frame through which the node received this GTK from the sender of the current frame;

– SSN if the current frame is secured by a PTK or GTK that has been used for one or more previously received frames with a valid MIC field, where SSN is the value of the Security Sequence Number field found in the last received frame containing a valid MIC and secured by this PTK or GTK.

The recipient shall discard all detected replayed frames.

Annex A (normative) Security keys

This annex provides the mechanisms for security key generation, exchange, and retirement. It contains two major sections, one on creating temporal keys, for frame authentication and encryption, between a node and a hub from their shared master key, and the other on setting out (i.e., activating) or setting up (i.e., establishing) a shared master key, for the creation of temporal keys, between the node and the hub by authenticated or unauthenticated association. The relationships between security keys and message securities are shown in Figure 4.

The cipher-based message authentication code algorithm (CMAC) as specified in the NIST Special Publication 800-38B, with the AES forward cipher function under a 128-bit key as specified in FIPS Pub 197, is used to compute keyed message authentication codes (KMAC) and derive keys needed in the key generation procedures described in this annex.

Specifically, the functional notation *CMAC(K, M)* represents the 128-bit output of the CMAC applied under key *K* to message *M* based on the AES forward cipher function.

Moreover, the bit string truncation functions *LMB_n(S)* and *RMB_n(S)* designate the *n* leftmost and the *n* rightmost bits of the bit string *S*, respectively. The sign || denotes concatenation of bit strings that are converted according to FIPS Pub 180-2 from certain fields of the frames of concern.

A.1 Temporal keys

A node and a hub shall jointly create a 128-bit secret pairwise temporal key (PTK) based on a 128-bit shared master key (MK) through PTK frames in order to exchange secured frames with each other. The node and the hub may create a new PTK based on their shared MK in preparation for the retirement of the current PTK. Either the node or the hub may initiate the procedure to create a PTK, if the two parties are not in the middle of creating another PTK between.

A hub shall choose a 128-bit secret group temporal key (GTK) and distribute it through unicast GTK frames to the nodes to which it desires to broadcast or multicast secured frames.

A.1.1 PTK creation

To initiate a PTK creation procedure, a node or a hub—referred to as the initiator—shall transmit the first PTK frame of the procedure to the intended recipient—referred to as the responder.

To continue the PTK creation procedure, the responder shall transmit the second PTK frame of the procedure, setting the PTK_KMAC field of the frame payload as depicted in Figure 22 to *PTK_KMAC_2* as calculated below. The responder shall set the PTK_KMAC field to 0 if it does not have a shared MK with the initiator.

To complete the PTK creation procedure, the initiator shall send the third PTK frame of the procedure, setting the PTK_KMAC field of the frame payload as depicted in Figure 22 to *PTK_KMAC_3* as also calculated below. The initiator shall send this PTK frame only after it has received the second PTK frame with the PTK_KMAC field set to *PTK_KMAC_2*.

Upon successfully sending the third PTK frame, the initiator shall compute a new PTK as given below, treating the responder's true identity as authenticated and the PTK creation procedure as completed. Upon receiving the third PTK frame with the PTK_KMAC field set to *PTK_KMAC_3*, the responder shall also compute the new PTK, treating the initiator's true identity authenticated and the PTK creation procedure completed as well.

The initiator and the responder shall independently generate a new 128-bit cryptographic random number as their Sender Nonce in a PTK creation procedure.

The initiator and the responder shall each derive the *PTK*, *KCK*, *PTK_KMAC_2*, and *PTK_KMAC_3* as follows:

PTK = CMAC(MK, Address_I || Address_R || Nonce_I || Nonce_R || PTK_Index)

KCK = CMAC(MK, Address_R || Address_I || Nonce_R || Nonce_I || PTK_Index)

P = CMAC(KCK, Address_I || Address_R || Nonce_R || Nonce_I || PTK_Index)

PTK_KMAC_2 = LMB_64(P), PTK_KMAC_3 = RMB_64(P)

The fields that form the message of CMAC correspond to the fields in the PTK frames of the current PTK creation procedure and are converted to bit strings according to FIPS Pub 180-2:

– *Address_I* is the Sender Address field of the frame payload of the first PTK frame.

– *Address_R* is the Recipient Address field of the frame payload of the first PTK frame.

– *Nonce_I* is the Sender Nonce field of the frame payload of the first PTK frame.

– *Nonce_R* is the Sender Nonce field of the frame payload of the second PTK frame.

– *PTK_Index* is the PTK Index field of the frame payload of the first PTK frame.

The PTK creation procedure is illustrated in Figure 56.

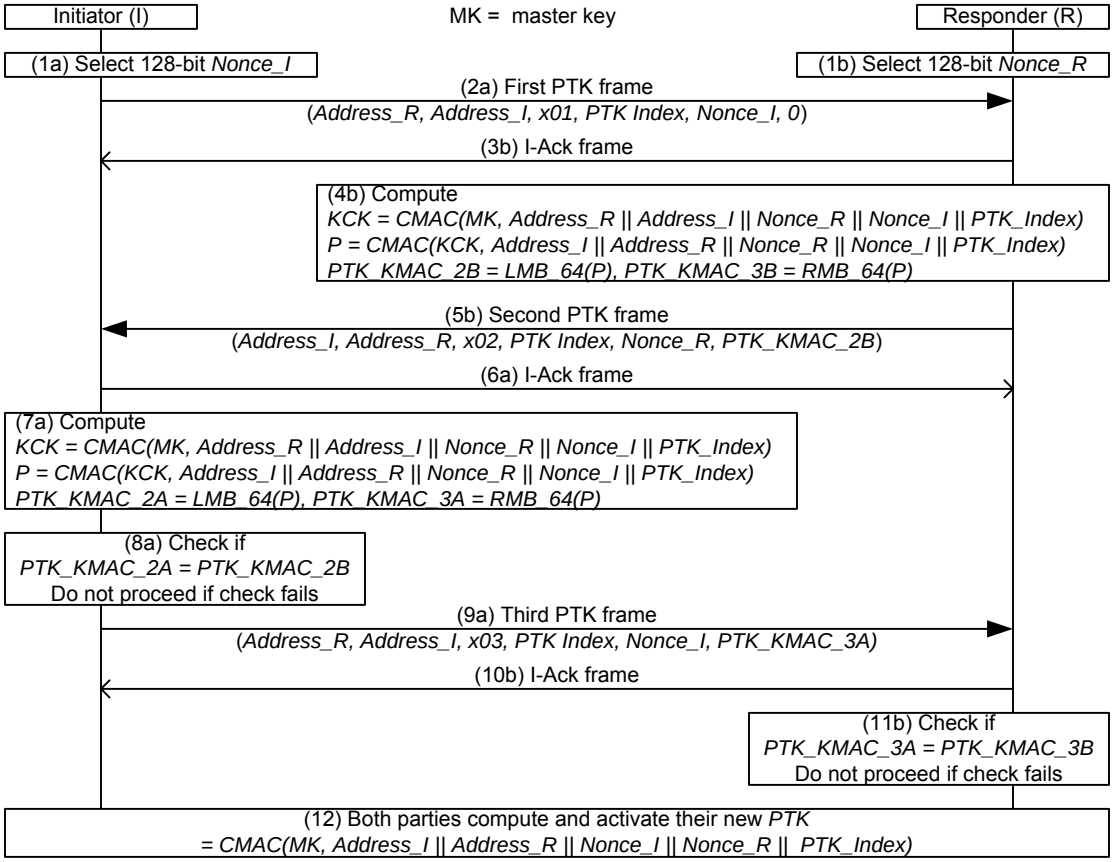


Figure 56 — PTK creation procedure

A.1.2 GTK distribution

A hub shall independently choose a new 128-bit cryptographic random number as the *GTK* that it needs to distribute to a group of nodes for broadcast or multicast of secured frames to that group. The hub shall send a GTK frame containing the *GTK* to each of the nodes in the group separately, with the GTK frame secured with the *PTK* being used between the hub and the intended recipient of the frame.

A.2 Master keys

A node and a hub shall have a 128-bit secret shared master key (MK) pre-shared or established jointly to create their pairwise temporal keys (PTK) for exchanging secured frames with each other. The node and the hub shall run one of the association protocols through exchange of Association frames as described below to activate a pre-shared MK or generate a new MK. To repeal an existing association and hence the shared MK, the node or the hub shall send a Disassociation frame.

The node—also referred to as the initiator—shall initiate an association procedure based on one of the association protocols supported by the hub as indicated in the Security Capability field of the hub's latest beacon. The hub—also referred to as the responder—shall respond to, but not initiate, an association procedure.

The node and the hub shall each have a secret pre-shared MK to run the MK pre-shared association protocol to activate their pre-shared MK as their shared MK for their PTK creation, with the benefit of keeping third parties not possessing the secret MK from launching impersonation attacks.

The node and the hub shall each require no authentication credentials such as a shared secret or human intervention to run the unauthenticated association protocol to generate their shared MK, without the benefit of keeping third parties from launching impersonation attacks.

The node and the hub shall have a secret transfer of the node's public key to the hub, typically through an out-of-band channel, to run the public key hidden association protocol to generate their shared MK, with the benefit of keeping third parties not possessing the node's secret private and public keys from launching impersonation attacks.

The node and the hub shall each have a secret shared password to run the password authenticated association protocol to generate their shared MK for their PTK creation, with the benefit of keeping third parties not possessing the secret password from launching impersonation attacks.

The node and the hub shall each have a display of a 5-digit decimal number to run the display authenticated association protocol to generate their shared MK for their PTK creation, with the benefit of keeping third parties not displaying the same generated secret from launching impersonation attacks.

For the unauthenticated association, public key hidden association, password authenticated association, and display authenticated association protocols, the node and the hub shall independently generate a new 128-bit cryptographic random number as their Sender Nonce in an association procedure. The node and the hub shall use the Diffie-Hellman key exchange based on the elliptic curve public key cryptography to derive their shared MK. The elliptic curve, characterized by the following equation

$$y^2 = x^3 + ax + b \ (\text{mod } p), \ a, b \in GF(p), \ 4a^3 + 27b^2 \neq 0$$

where $GF(p)$ is a prime finite field, shall have the following values for its coefficients and domain parameters, as specified for Curve P-192 in FIPS Pub186-2, with p (an odd prime), r (order of base point G), and a (a coefficient) given in decimal form, and coefficient b and base point $G = (G_x, G_y)$ given in hex:

p = `6277101735386680763835789423207666416083908700390324961279`

r = `6277101735386680763835789423176059013767194773182842284081`

$a = -3 \ \text{mod } p$

b = `64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1`

G_x = `188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012`

G_y = `07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811`

The private keys (also called secret keys) SK_A and SK_B of the elliptic curve public key cryptography for the node and the hub, respectively, shall be chosen as unique 192-bit integers in the range $[1, r-1]$. The corresponding 192-bit public keys PK_A and PK_B shall be computed as follows:

$$PK_A = SK_A \times G, \ PK_B = SK_B \times G$$

where × denotes scalar multiplication of the base point *G* by an integer as described in A.9.2 of IEEE Std P1363-2000. A received public key, denoted by an *X*-coordinate value or a pair of *X*-coordinate and *Y*-coordinate values, shall be treated valid only if it is a non-infinity point on the elliptic curve defined in the above, i.e., that its *X* and *Y* coordinates shall satisfy the elliptic curve equation given above.

A.2.1 Master key (MK) pre-shared association

To initiate a procedure for the MK pre-shared association protocol, a node shall transmit the first Association frame of the procedure.

To continue the association procedure, the recipient hub shall transmit the second Association frame of the procedure. If the hub does not have a pre-shared MK shared with this node, it shall set the Association Protocol Number field of the frame payload to a value indicating a different association protocol that the node may use to associate with the hub.

Upon successfully sending the second Association frame without changing the Association Protocol Number, the hub shall activate the pre-shared MK as their shared MK, treating the node's true identity as unauthenticated but the association procedure as completed. Upon receiving the second Association frame with the same Association Protocol Number as contained in the first one, the node shall also activate the pre-shared MK as their shared MK, treating the hub's true identity unauthenticated but the association procedure completed as well. The node shall proceed to the PTK creation procedure to create a PTK with the hub, meanwhile performing mutual authentication of each other based on the claimed pre-shared MK.

The MK pre-shared association procedure is illustrated in Figure 57.

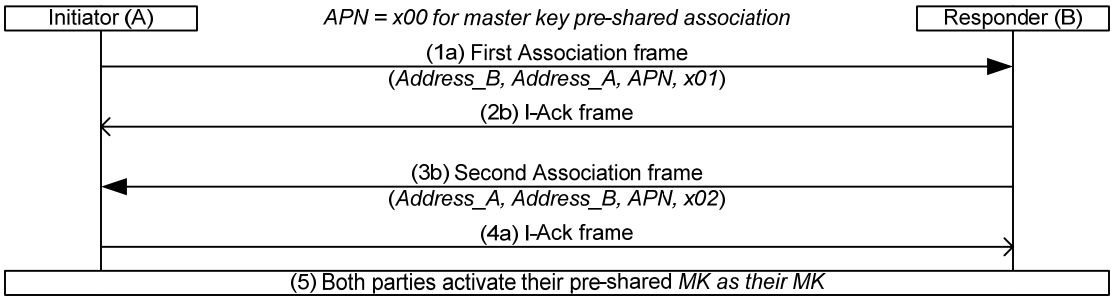


Figure 57 — MK pre-shared association procedure

A.2.2 Unauthenticated association

To initiate a procedure for the unauthenticated association protocol, a node shall transmit the first Association frame of the procedure.

To continue the association procedure, the recipient hub shall transmit the second Association frame of the procedure, setting the MK_KMAC field of the Association Data as depicted in Figure 20 to *MK_KMAC_2* as calculated below. If the hub does not accept unauthenticated association with this node, it shall set the Association Protocol Number field of the frame payload to a value indicating a different association protocol that the node may use to associate with the hub.

To continue further the association procedure, the node shall send the third Association frame of the procedure, setting the MK_KMAC field of the Association Data as depicted in Figure 20 to *MK_KMAC_3* as also calculated below. The node shall send this Association frame only after it has received the second Association frame with the MK_KMAC field set to *MK_KMAC_2*.

Upon successfully sending the third Association frame, the node shall compute the shared MK as given below, treating the hub's true identity as unauthenticated but the association procedure as completed. Upon receiving the third Association frame with the MK_KMAC field set to *MK_KMAC_3*, the hub shall also compute the shared MK as given below, treating the node's true identity unauthenticated but the association procedure completed as well.

The node and the hub shall each compute a DHKey as follows:

$DHKey = X(SK_A \times PK_B) = X(SK_B \times PK_A) = X(SK_A \times SK_B \times G)$

The node and the hub shall each derive *MK_KMAC_2* and *MK_KMAC_3* as follows:

MK_KMAC_2 = CMAC(DHKey, Address_A || Address_B || Nonce_A || Nonce_B)

MK_KMAC_3 = CMAC(DHKey, Address_B || Address_A || Nonce_B || Nonce_A)

The node and the hub shall each derive their shared MK as follows:

MK = CMAC(DHKey, Nonce_A || Nonce_B)

In the above, $X(P) = X(P_X, P_Y) = P_X$ = *X*-coordinate of *P*, which is computed from $SK_A \times PK_B$ at the node and from $SK_B \times PK_A$ at the hub, respectively.

- SK_A is the node's 192-bit private key (an integer) kept secret by the node.
- SK_B is the hub's 192-bit private key kept secret by the hub.
- PK_A is the node's 192-bit public key (a pair of *X* and *Y* coordinates) transmitted by the node.
- PK_B is the hub's 192-bit public key (a pair of *X* and *Y* coordinates) transmitted by the hub.
- *Address_A* is the Sender Address field of the frame payload of the first Association frame.
- *Address_B* is the Recipient Address field of the frame payload of the first Association frame.
- *Nonce_A* is the Sender Nonce field of the frame payload of the first Association frame.
- *Nonce_B* is the Sender Nonce field of the frame payload of the second Association frame.

The unauthenticated association procedure is illustrated in Figure 58.

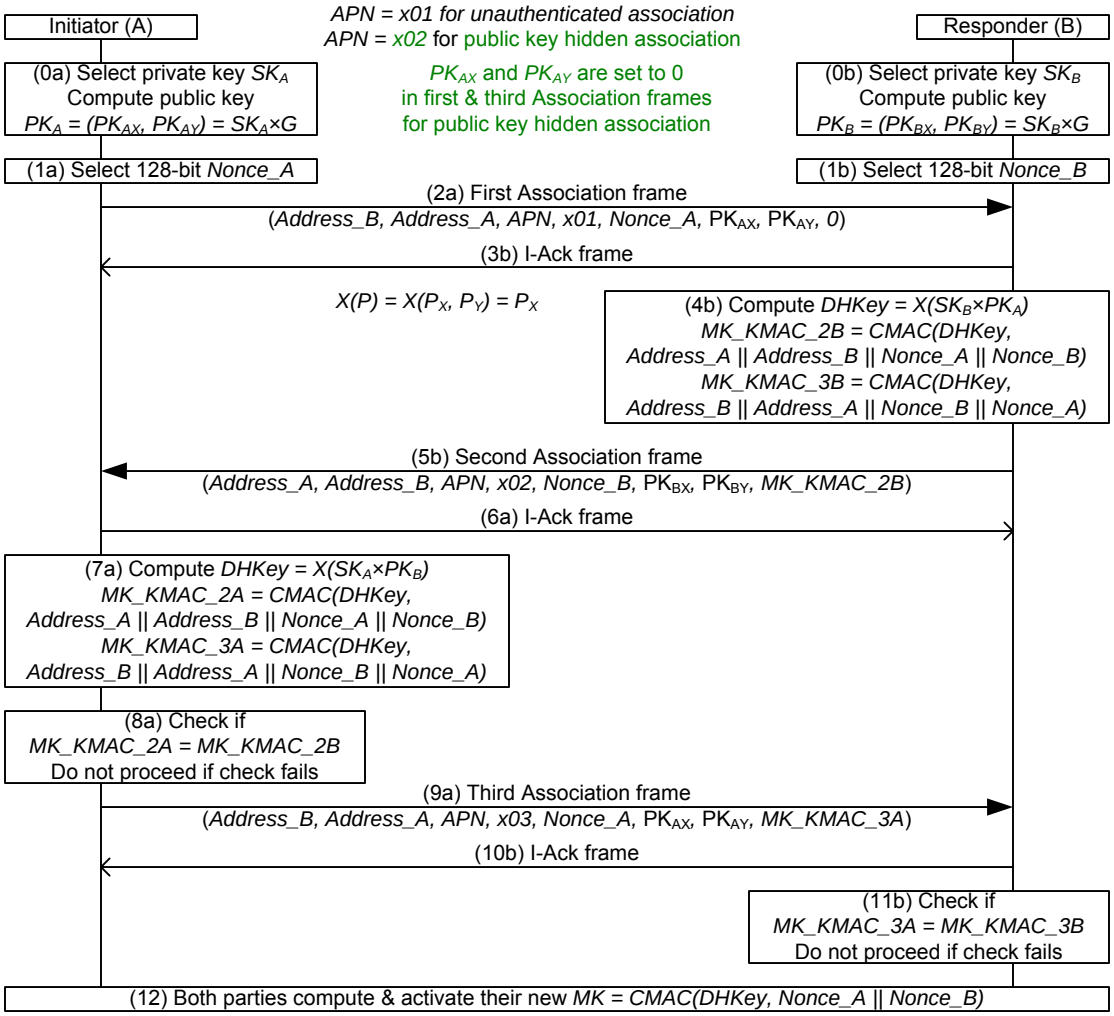


Figure 58 — Unauthenticated association procedure

A.2.3 Public key hidden association

To initiate a procedure for the public key hidden association protocol, a node shall transmit the first Association frame of the procedure.

To continue the association procedure, the recipient hub shall transmit the second Association frame of the procedure, setting the MK_KMAC field of the Association Data as depicted in Figure 20 to *MK_KMAC_2* as calculated below. If the hub does not have the public key of this node, it shall set the Association Protocol Number field of the frame payload to a value indicating a different association protocol that the node may use to associate with the hub, and shall set the MK_KMAC field to 0.

To continue further the association procedure, the node shall send the third Association frame of the procedure, setting the MK_KMAC field of the Association Data as depicted in Figure 20 to *MK_KMAC_3* as also calculated below. The node shall send this Association frame only after it has received the second Association frame with the MK_KMAC field set to *MK_KMAC_2*.

Upon successfully sending the third Association frame, the node shall compute the shared MK as given below, treating the hub's true identity as authenticated and the association procedure as completed. Upon receiving the third Association frame with the MK_KMAC field set to *MK_KMAC_3*, the hub shall also compute the shared MK as given below, treating the node's true identity authenticated and the association procedure completed as well.

The node and the hub shall each compute a DHKey as follows:

$DHKey = X(SK_A \times PK_B) = X(SK_B \times PK_A) = X(SK_A \times SK_B \times G)$

The node and the hub shall each derive *MK_KMAC_2* and *MK_KMAC_3* as follows:

MK_KMAC_2 = CMAC(DHKey, Address_A || Address_B || Nonce_A || Nonce_B)

MK_KMAC_3 = CMAC(DHKey, Address_B || Address_A || Nonce_B || Nonce_A)

The node and the hub shall each derive their shared MK as follows:

MK = CMAC(DHKey, Nonce_A || Nonce_B)

In the above, $X(P) = X(P_X,P_Y) = P_X$ = *X*-coordinate of *P*, which is computed from $SK_A \times PK_B$ at the node and from $SK_B \times PK_A$ at the hub, respectively.

– SK_A is the node's 192-bit private key (an integer) kept secret by the node.

– SK_B is the hub's 192-bit private key kept secret by the hub.

– PK_A is the node's 192-bit public key (a pair of *X* and *Y* coordinates) transferred only to the hub by a secure out-of-band channel.

– PK_B is the hub's 192-bit public key (a pair of *X* and *Y* coordinates) transmitted by the hub.

– *Address_A* is the Sender Address field of the frame payload of the first Association frame.

– *Address_B* is the Recipient Address field of the frame payload of the first Association frame.

– *Nonce_A* is the Sender Nonce field of the frame payload of the first Association frame.

– *Nonce_B* is the Sender Nonce field of the frame payload of the second Association frame.

The public key hidden association procedure is also illustrated in Figure 58.

A.2.4 Password authenticated association

To initiate a procedure for the password authenticated association protocol, a node shall transmit the first Association frame of the procedure, setting the Sender PK_X and Sender PK_Y fields of the Association Data as depicted in Figure 20 to the *X*-coordinate PK'_{AX} and *Y*-coordinate PK'_{AY}, respectively, of the node's password-scrambled public key PK'_A as calculated below.

To continue the association procedure, the recipient hub shall transmit the second Association frame of the procedure, setting the MK_KMAC field of the Association Data to *MK_KMAC_2* as calculated below. If the hub does not have a shared password with this node, it shall set the Association Protocol Number field of the frame payload to a value indicating a different association protocol that the node may use to associate with the hub, and shall set the MK_KMAC field to 0.

To continue further the association procedure, the node shall send the third Association frame of the procedure, setting the Sender PK_X and Sender PK_Y fields of the Association Data as depicted in Figure 20 to the values of the corresponding fields of the Association Data in the first Association frame of the procedure, and setting the MK_KMAC field of the Association Data to *MK_KMAC_3* as also calculated below. The node shall send this Association frame only after it has received the second Association frame with the MK_KMAC field set to *MK_KMAC_2*.

Upon successfully sending the third Association frame, the node shall compute the shared MK as given below, treating the hub's true identity as authenticated and the association procedure as completed. Upon receiving the third Association frame with the MK_KMAC field set to *MK_KMAC_3*, the hub shall also compute the shared MK as given below, treating the hub's true identity authenticated and the association procedure completed as well.

The node shall compute its password-scrambled public key $PK'_A = (PK'_{AX}, PK'_{AY})$ from its public or private key and the password shared with the hub as follows:

$PK'_A = PK_A - Q(PW) = SK_A \times G - Q(PW)$

$Q(PW) = (Q_X = PW+M_X, Q_Y = \text{even positive integer})$

The hub shall recover the node's public key from the received password-scrambled public key $PK'_A = (PK'_{AX}, PK'_{AY})$ for the subsequent *DHKey* computation as follows:

$PK_A = PK'_A + Q(PW), \quad Q(PW) = (Q_X = PW+M_X, Q_Y = \text{even positive integer})$

The parameters involved in these equations are defined below:

– PW is a positive integer converted according to IEEE Std P1363-2000 from the UTE-16BE representation of the shared password by treating the leftmost octet as the octet containing the most-significant bits.

– M_X is the smallest nonnegative integer such that $Q_X = PW+M_X$ is the X-coordinate of a point on the elliptic curve defined earlier.

– $Q(PW)$ is the point on the elliptic curve with X-coordinate = Q_X and Y-coordinate = Q_Y of an even positive integer.

The node shall choose a private key SK_A such that the X-coordinate of PK_A is not equal to the X-coordinate of $Q(PW)$. If the node first chose a private key SK_A such that the X-coordinate of PK_A was equal to the X-coordinate of $Q(PW)$, it may update SK_A and PK_A as follows: $SK_A := (1 + SK_A) \bmod p$, $PK_A := PK_A + G$, where p and G are defined earlier in A.2.

The node and the hub shall each compute a DHKey as follows:

$DHKey = X(SK_A \times PK_B) = X(SK_B \times PK_A) = X(SK_A \times SK_B \times G)$

The node and the hub shall each derive *MK_KMAC_2* and *MK_KMAC_3* as follows:

MK_KMAC_2 = CMAC(DHKey, Address_A || Address_B || Nonce_A || Nonce_B)

MK_KMAC_3 = CMAC(DHKey, Address_B || Address_A || Nonce_B || Nonce_A)

The node and the hub shall each derive their shared MK as follows:

MK = CMAC(DHKey, Nonce_A || Nonce_B)

In the above, $X(P) = X(P_X,P_Y) = P_X = X$-coordinate of P, which is computed from $SK_B \times PK_A$ at the hub and from $SK_A \times PK_B$ at the node, respectively.

– SK_A is the node's 192-bit private key (an integer) kept secret by the node.

– SK_B is the hub's 192-bit private key kept secret by the hub.

– PK_A is the node's 192-bit public key (a pair of X and Y coordinates) kept secret by the node.

– PK_B is the hub's 192-bit public key (a pair of X and Y coordinates) transmitted by the hub.

– *Address_A* is the Sender Address field of the frame payload of the first Association frame.

– *Address_B* is the Recipient Address field of the frame payload of the first Association frame.

– *Nonce_A* is the Sender Nonce field of the frame payload of the first Association frame.

– *Nonce_B* is the Sender Nonce field of the frame payload of the second Association frame.

The password authenticated association procedure is illustrated in Figure 59.

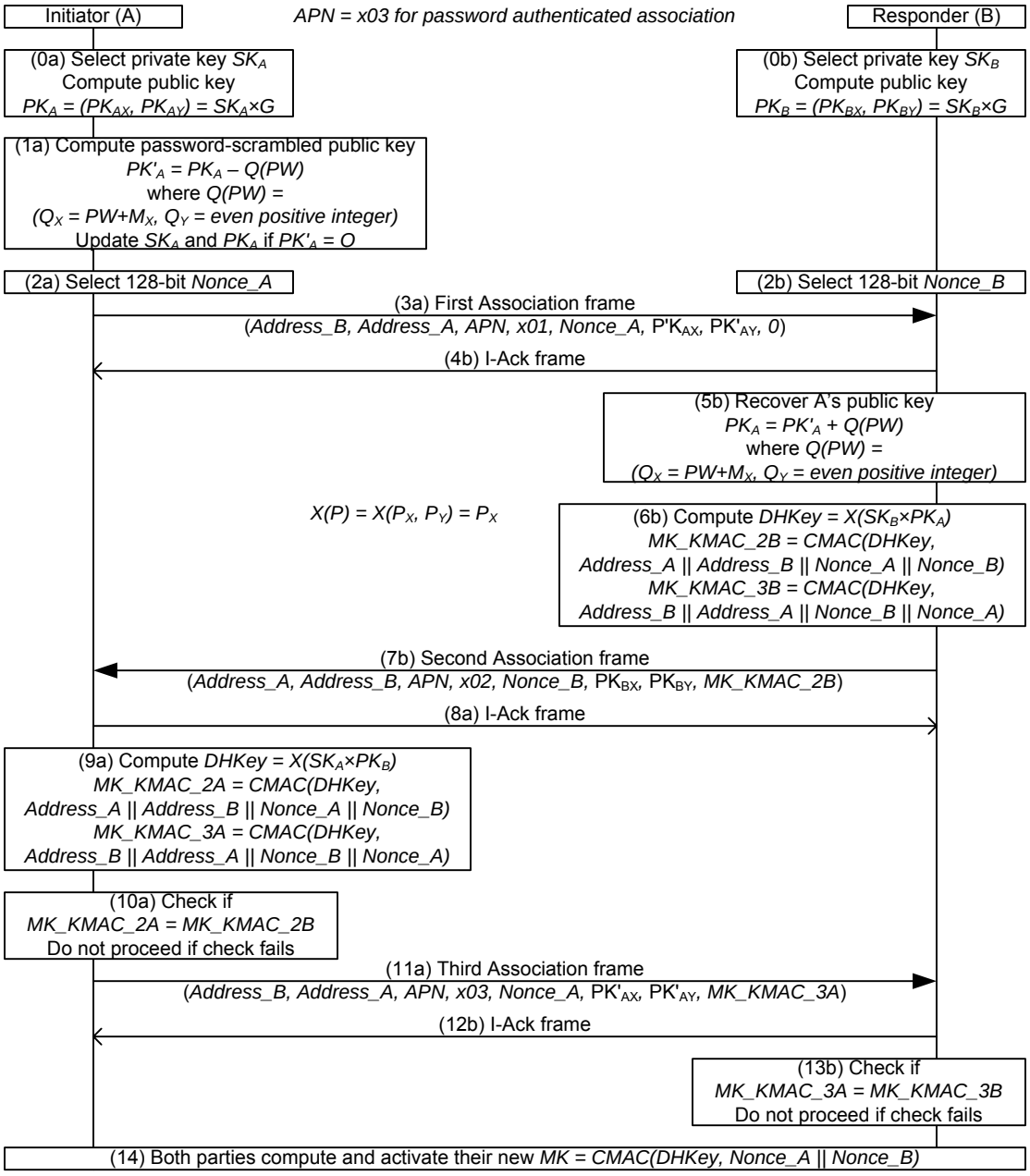


Figure 59 — Password authenticated association procedure

A.2.5 Display authenticated association

To initiate a procedure for the display authenticated association protocol, a node shall transmit the first Association frame of the procedure, setting the MK_KMAC field of the Association Data as depicted in Figure 20 to *Commitment* as calculated below.

To continue the association procedure, the recipient hub shall transmit the second Association frame of the procedure. If the hub does not accept display authenticated association with this node, it shall set the Association Protocol Number field of the frame payload to a value indicating a different association protocol that the node may use to associate with the hub.

To continue further the association procedure, the node shall send the third Association frame of the procedure. The hub shall not treat this frame as valid and shall display a number of 0 if the MK_KMAC field contained in the received first Association frame of the procedure is not equal to *Commitment* as given below.

The node shall display a 5-digit decimal number *Display_A* immediately after successfully sending the third Association frame, and the hub shall also display a 5-digit decimal number *Display_B* immediately after determining the received third Association frame to be valid. If the node and the hub display the same 5-digit number, they shall each be informed through their respective user interfaces that their mutual authentication has succeeded. Otherwise, they shall each be informed that their mutual authentication has failed.

Upon determining that their mutual authentication has succeeded, the node and the hub shall each compute the shared MK as given below, treating their authenticated association procedure as completed.

The node and the hub shall each compute a DHKey as follows:

$DHKey = X(SK_A \times PK_B) = X(SK_B \times PK_A) = X(SK_A \times SK_B \times G)$

The node and the hub shall each derive *Commitment* as follows:

$Commitment = CMAC(Nonce\_A, Address\_A \,||\, Address\_B \,||\, PK_{AX} \,||\, PK_{AY})$

The node and the hub shall also compute *Display_A* and *Display_B*, respectively, as follows:

$H = CMAC(DHKey, Address\_A \,||\, Address\_B \,||\, Nonce\_A \,||\, Nonce\_B,,=), \quad D = RMB\_16(H)$

$Display\_A = BS2DI(D), \quad Display\_B = BS2DI(D)$

The node and the hub shall each derive their shared MK as follows:

$MK = CMAC(DHKey, Nonce\_A \,||\, Nonce\_B)$

In the above, $X(P) = X(P_X,P_Y) = P_X = X$-coordinate of P, which is computed from $SK_A \times PK_B$ at the node and from $SK_B \times PK_A$ at the hub, respectively. *BS2DI(BS)* converts the bit string *BS*, based on IEEE Std P1363-2000, to a positive decimal integer by treating the leftmost bit of the string as the most-significant bit of the integer.

- SK_A is the node's 192-bit private key (an integer) kept secret by the node.
- SK_B is the hub's 192-bit private key kept secret by the hub.
- PK_A is the node's 192-bit public key (a pair of X and Y coordinates) transmitted by the node in the third Association frame.
- PK_B is the hub's 192-bit public key (a pair of X and Y coordinates) transmitted by the hub in the second Association frame.
- PK_{AX} and PK_{AY} are the X-coordinate and Y-coordinate, respectively, of the node's 192-bit public key (a pair of X and Y coordinates) contained in the third Association frame.
- *Address_A* is the Sender Address field of the frame payload of the third Association frame.
- *Address_B* is the Recipient Address field of the frame payload of the third Association frame.
- *Nonce_A* is the Sender Nonce field of the frame payload of the third Association frame.
- *Nonce_B* is the Sender Nonce field of the frame payload of the second Association frame.

The display authenticated association procedure is illustrated in Figure 60.

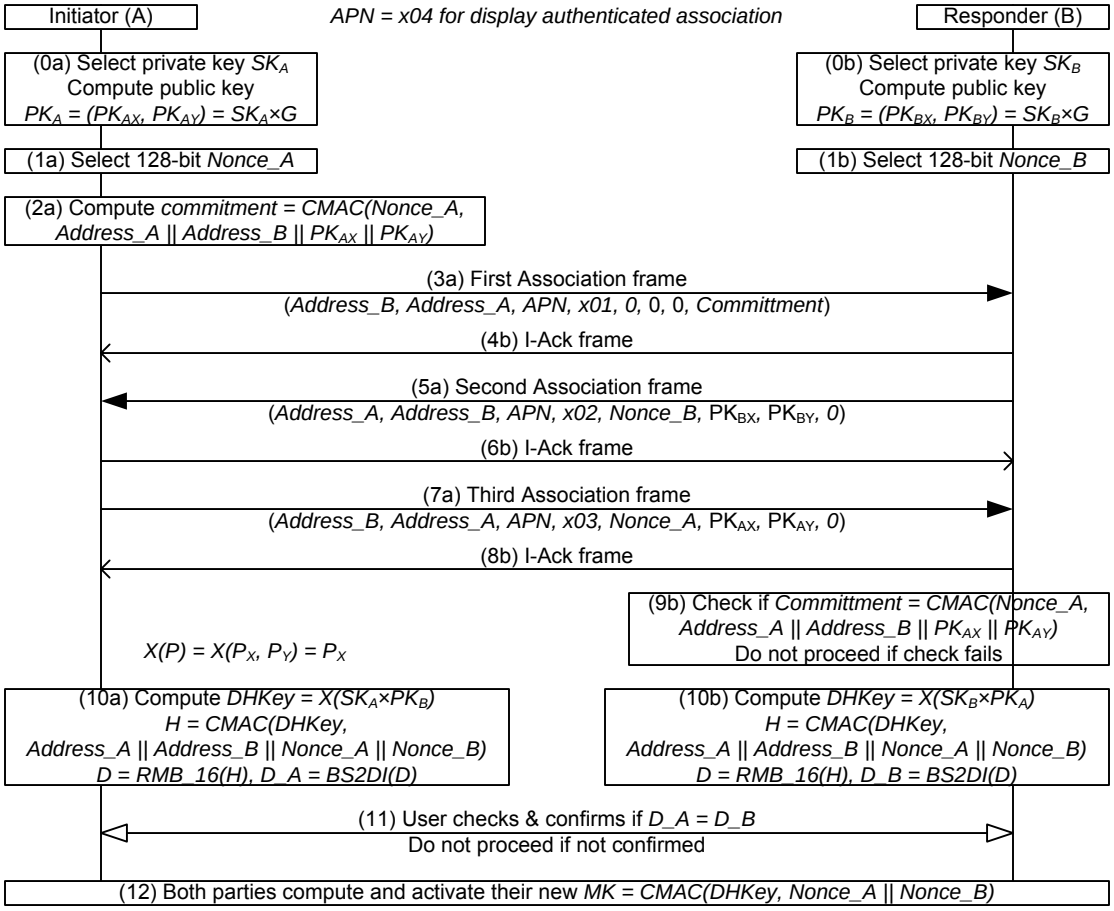


Figure 60 — Display authenticated association procedure

A.2.6 Disassociation

To initiate a disassociation procedure for nullifying an existing association and hence the shared MK and PTK with a hub or a node, the node or the hub shall send a Disassociation frame, setting the DA_KMAC field of the frame payload as depicted in Figure 21 to *DA_KMAC*. Upon successfully sending the Disassociation frame, the sender shall erase the MK and the corresponding PTK materials from its internal storage.

Upon receiving a Disassociation frame with the DA_KMAC field set to *DA_KMAC*, the recipient shall also erase the MK and the corresponding PTK materials from its internal storage.

To send a Disassociation, the sender shall independently generate a new 128-bit cryptographic random number as its Sender Nonce in the Disassociation frame.

The node and the hub shall compute *DA_KMAC* as follows:

DA_KMAC = CMAC(MK, Address_A || Address_B || Nonce_A)

The input fields to the computation above are defined as follows:

– *MK* is the shared MK to be repealed.

– *Address_A* is the Sender Address field of the frame payload of the Disassociation frame.

– *Address_B* is the Recipient Address field of the frame payload of theDisassociation frame.

– *Nonce_A* is the Sender Nonce field of the frame payload of the Disassociation frame.

The disassociation procedure is illustrated in Figure 61.

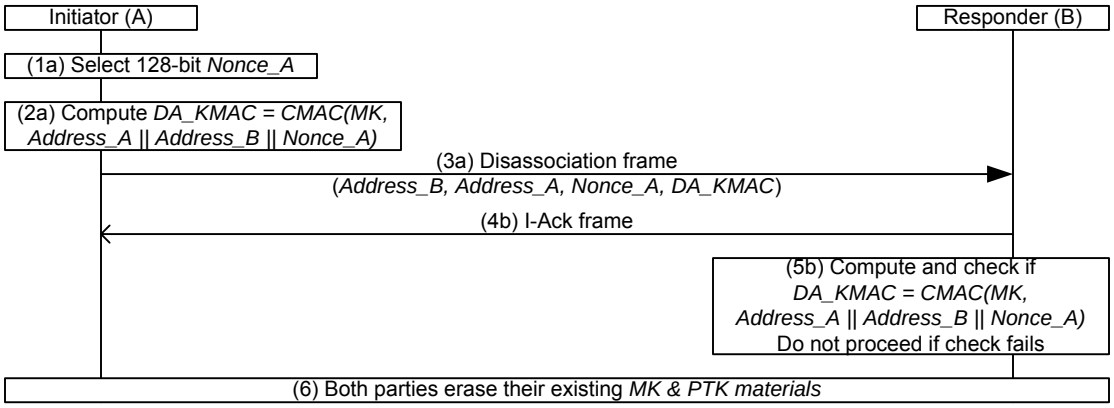


Figure 61 — Disassociation procedure