STANDARD

HART COMMUNICATION PROTOCOL

TDMA Data Link Layer Specification

HCF_SPEC-075, Revision 1.1

Release Date: 17 May, 2008

Release Date: 17 May, 2008

Document Distribution / Maintenance Control / Document Approval
To obtain information concerning document distribution control, maintenance control, and document approval please contact the HART Communication Foundation (HCF) at the address shown below.

Trademark Information
HART® is a registered trademark of the HART Communication Foundation, Austin, Texas, USA. Any use of the term HART hereafter in this document, or in any document referenced by this document, implies the registered trademark. WirelessHART™ is a trademark of the HART Communication Foundation. All other trademarks used in this or referenced documents are trademarks of their respective companies. For more information contact the HCF Staff at the address below.

Attention: Foundation Director
HART Communication Foundation
9390 Research Boulevard
Suite I-350
Austin, TX 78759, USA
Voice: (512) 794-0369
FAX: (512) 794-3904

http://www.hartcomm.org

Intellectual Property Rights
The HCF does not knowingly use or incorporate any information or data into the HART Protocol Standards which the HCF does not own or have lawful rights to use. Should the HCF receive any notification regarding the existence of any conflicting Private IPR, the HCF will review the disclosure and either (a) determine there is no conflict; (b) resolve the conflict with the IPR owner; or (c) modify the standard to remove the conflicting requirement. In no case does the HCF encourage implementers to infringe on any individual's or organization's IPR.

9390 Research Blvd., Ste. I-350
Austin, TX 78759 USA
Tel: 512-794-0369
Fax: 512-794-3904

Addendum To
TDMA Data-Link Layer Specification (HCF_SPEC-075)

May 17, 2008

Since release of the HART Communication Protocol Revision 7.0 Specifications in September 2007, HCF staff and Technical Working Groups updating and developing test specifications have closely scrutinized the Protocol Specifications. Any anomalies, errors or omissions discovered in the Specifications have been identified, tracked and resolved. Changes, clarifications and corrections resulting from the anomalies discovered and resolved during this process are detailed in this addendum.

This addendum provides developers with the most current, accurate and up-to-date information on the HART 7 Specifications. Each change is detailed below by Subsection and brief explanation of the change. All changes described in this addendum are mandatory. HART-enabled product implementations must comply with the Specification corrections and clarifications described in this addendum.

Subsection 7.2/Table1

In Table 1 modify the `READ_MAX_BACK_OFF_EXPONENT` `ADD_CONNECTION` *and* `DELETE_CONNECTION` *services as follows:*

READ_MAX_BACK_OFF_EXPONENT		The maximum value that can be assumed for the back-off exponent used in shared slots. Valid values are {4, 5, 6, 7}. MaxBackoffExponent defaults to 4.
	Unsigned-4 MaxBackoffExponent	

ADD_CONNECTION		Adds a new connection to a device via a specified graph
	Unsigned-16 connectionHandle	Handle for this connection
	Unsigned-16 graphId	graphId for the connection
	Unsigned-16 nodeNickname	address of device being connected via the specified graph
DELETE_CONNECTION		Deletes an existing connection
	Unsigned-16 connectionHandle	Handle of an existing connection

Subsection 8.2.4:

The Transmit bit in advertise DLPDUs is with respect to the joining devices. In other words, when the link transmit, bit 6, is set the joining devices may use the link to propagate a join request to the Network Manager. Consequently, paragraph 3 of Subsection 8.2.4 must be modified to read:

Once the basic network information is disclosed, the Advertise DLPDU lists all of its join links by superframe. In addition, each link is identified as either transmit or receive from the perspective of the joining device. The joining device shall assume all . . .

And the Link specification modified as follows:

For each Link

Unsigned-16	Join slot. The specific slot within the superframe for this Link.
Bits	Join slot channel offset.
x.7	Reserved. Must be set to zero. No device shall make any assumption regarding their possible future use of this bit.
x.6	When set, the link is for DLPDU transmission by the joining device.
x.5-x.0	Channel offset. The frequency channel offset for this slot. This value is used to calculate the link frequency/channel.

Subsection 9.2.2

The link channel calculation uses an ordered list of monotonically increasing channel numbers. To make this more clear in the specification the last two paragraphs in Subsection 9.2.2 must be modified to read as follows:

The bit number of the least significant active channel is placed at index zero in the array. The bit number corresponds to the index into the Physical Channel Table for the Physical Layer in use. The ActiveChannel value is used to index into the ActiveChannelArray.

Channel = ActiveChannelArray [ActiveChannel]

The result is the channel that must be used for communication in that Absolute Slot Number. The ActiveChannelArray is an ordered list beginning with the smallest active channel index (at ActiveChannelArray [0]) through the largest.

Subsection 9.2.3

Details on averaging the RSL must be clarified. After paragraph 4 these details must be inserted as follows:

The device's ability to communicate with a neighbor is a key metric in forming and grooming the mesh network. Consequently, statistics are maintained in each neighbor table entry. These include average Received Signal Level (RSL); statistics on the packets transmitted and received and the timestamp of the last communication with the neighbor.

For linked neighbors, RSL is calculated using an IIR filter using the following equation:

$$\text{RSL} = \text{RSL} - (\text{RSL} / \text{RSLDamp}) + (\text{MeasuredRSL} / \text{RSLDamp})$$

Where MeasuredRSL is the RSL for the current packet and RSLDamp is the damping factor. RSLDamp must be a power of 2 and defaults to 64. For discovered or un-linked neighbors (i.e., neighbors the device does not communicate with) the highest RSL value is returned.

If a link to that neighbor exists, the LastTimeCommunicated is used to trigger transmission of Keep-Alive packets. A Keep-Alive must be transmitted to the neighbor (see Subsection 9.3) whenever the LastTimeCommunicated is greater than the keepAliveInterval. Keep-Alive transmissions are repeated until a new DLPDU is received from the neighbor.

In Table 7 of Subsection 9.2.3, BOCntr must be increased in size from 5 bits to 8 bits. The corresponding entry in Table 7 must be modified as follows:

Unsigned-8 BOCntr	Back-off countdown in collision avoidance algorithm for shared links

Subsection 9.2.4

The Graph ID field is overloaded combining frame-based routing, graph-based routing and "Invalid" Graph in the same fields. This must be clarified. Consequently, the first paragraph in Subsection 9.2.4 must be modified to read as follows:

The graph provides the routing information to guide the delivery of a packet to its final destination. A graph is a directed list of paths that connect two devices within the network. Both upstream (toward the Gateway) and downstream graphs are used in WirelessHART. The Network Manager is responsible for correctly configuring each graph. Graphs have an ID; a list of neighbors and (optionally) the destination's long and short address. When the Graph ID value is less than 0x0100 it indicates a Superframe ID, and when equal to 0xFFFF it indicates the Graph is Invalid.

Subsection 9.2.5

It is important that the highest priority, oldest packet is the first forwarded. The age of the packet must be judged based on the ASN Snippet not when the device received the packet. Consequently, The discussion of the PacketTimeStamp must be deleted from the text and Table 9. Subsection 9.2.5 is modified to read as follows:

All devices must maintain a list of packet buffers. These buffers are used to receive, process and transmit packets. The record associated with a packet is indicated in Table 9. The Packet ID is used to reference the packet and is created when the packet is added via the TRANSMIT.request service primitive.

The ASN Snippet (see the *Network Management Specification*) is used to select the packet to transmit when either of two equal priority packets can be propagated in the same absolute slot. In this case the older packet is sent first. Also, in worst-case scenarios after a long time elapses, by comparing to MaxPacketAge, the ASN Snippet is used to automatically flush a very old packet.

In addition, the record contains the packet's priority and the specification of the packet's destination.

Table 9. Packet Record

Content	Description
PacketId	Unique Packet ID.
Payload	The Data-Link Payload (i.e. the NPDU)
Priority	Transmit priority of the packet
PacketBirthASN	Determines the age of the packet (based on the NPDU ASNSnippet).
Destination	Graph, source or proxy routing information or broadcast destination.

Subsection 9.2.5/Table 10

It is important that the highest priority, oldest packet is the first forwarded. The age of the packet must be judged based on the ASN Snippet. Consequently, Table 10 is modified to read as follows:

Table 10. Packet Precedence Order

Tie-Breaker Number	Rule (Apply Top-to Bottom until a single packet is identified)
0	Choose the packet(s) with the highest priority
1	Choose the oldest packet based on the ASN Snippet (see the *Network Management Specification*)
2	Choose the packet destined to the neighbor communicated with longest ago.
3	If keep alive time has expired with a neighbor, generate a keep alive packet to the neighbor with communicated with longest ago.
4	If an advertise time has lapsed (see Subsection 9.3.4) then generate an Advertise packet.

Subsection 9.3.2

Unless there is a transmit link to be serviced in the same slot, all receive links must be serviced. This must be clarified in the Subsection 9.3.2. Consequently, Subsection 9.3.2 must be changed as follows:

For each active superframe, all receive links will be scheduled. The set of upcoming receive links can be calculated in the same fashion as with transmit links (see Subsection 9.3.1). The main difference is that, baring the need to propagate a packet, all receive links must be serviced. Once the ordered list of links is . . .

Subsection 9.3.3/Table11.

For clarity, Sets of BOCntr values for BOExp = 5, 6, 7 must be included in Table 11. Consequently Table 11 is enhanced as follows:

BOExp	Set of Possible Values for BOCntr
1	{0, 1}
2	{0, 1, 2, 3}
3	{0, 1, 2, 3, 4, 5, 6, 7}
4	{0, 1, 2, 3, 4, 5, . . . , 12, 13, 14, 15}
5	{0, 1, 2, 3, 4, 5, . . . , 28, 29, 30, 31}
6	{0, 1, 2, 3, 4, 5, . . . , 60, 61, 62, 63}
7	{0, 1, 2, 3, 4, 5, . . . , 124, 125, 126, 127}

Subsection 9.3.4

Transmitting a Keep-Alive is more important than advertising. The requirement to use an Advertisement DLPDU in place of a Keep-Alive is too complicated and must be stricken. Consequently, Subsection 9.3.4 is modified to read as follows:

Nodes that are already part of the network may be configured by the Network Manager to advertise the network and facilitate joining of new devices. The AdvertiseInterval attribute sets the interval at which Advertise packets (see Subsection 8.2.4) are generated. Whenever the AdvertiseInterval lapses an Advertise packet shall be transmitted on the first available non-shared transmit link. When AdvertiseInterval is set to zero then an Advertise packet shall be generated whenever a non-shared transmit link is available.

An Advertise packet may be sent on any non-shared transmit link that is not in use. All other traffic has higher priority than advertising.

Subsection 9.4.1/Figure 15:

The state transition from `Listen` *to* `Idle` *with the event:* `Message Discarded && (Destination == Broadcast)` *must be corrected. The correct event is simply* `Message Discarded`*. In addition another state transition must be added from from* `Listen` *to* `Idle` *with the event:* `Message Not Accepted && (Destination == Broadcast)` *(e.g., due to flow control in buffers)*

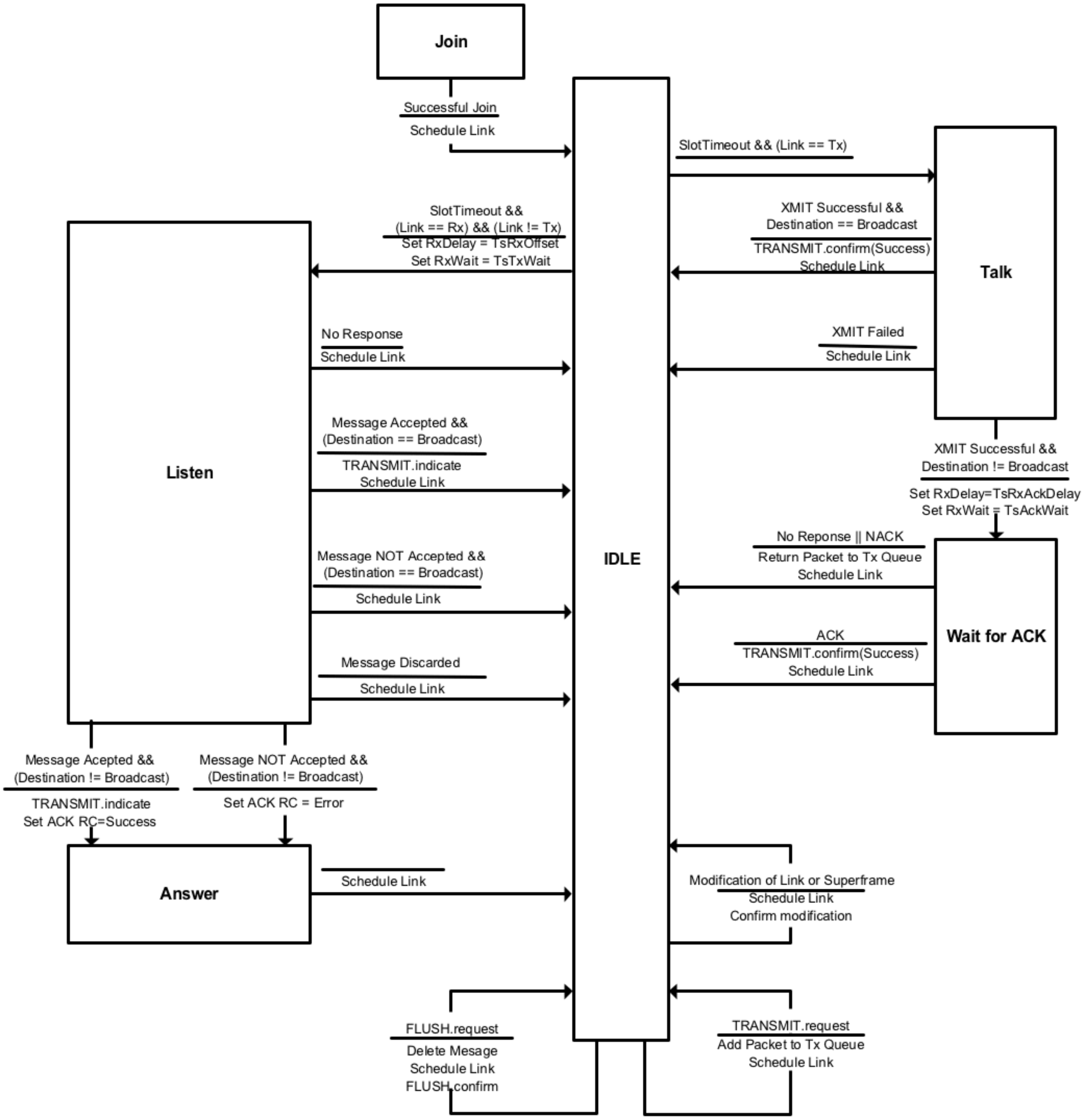


Figure 15. TDMA State Machine

Subsection 9.4.3/Figure 18

The Receive State Machine in Figure 18 must be corrected to indicate all valid receive packets (even if they are not addressed to the device) must update the neighbor table. Currently, the state machine (incorrectly) indicates a packet not addressed to the device is immediately discarded. In reality that packet must still be validated and the neighbor table updated accordingly

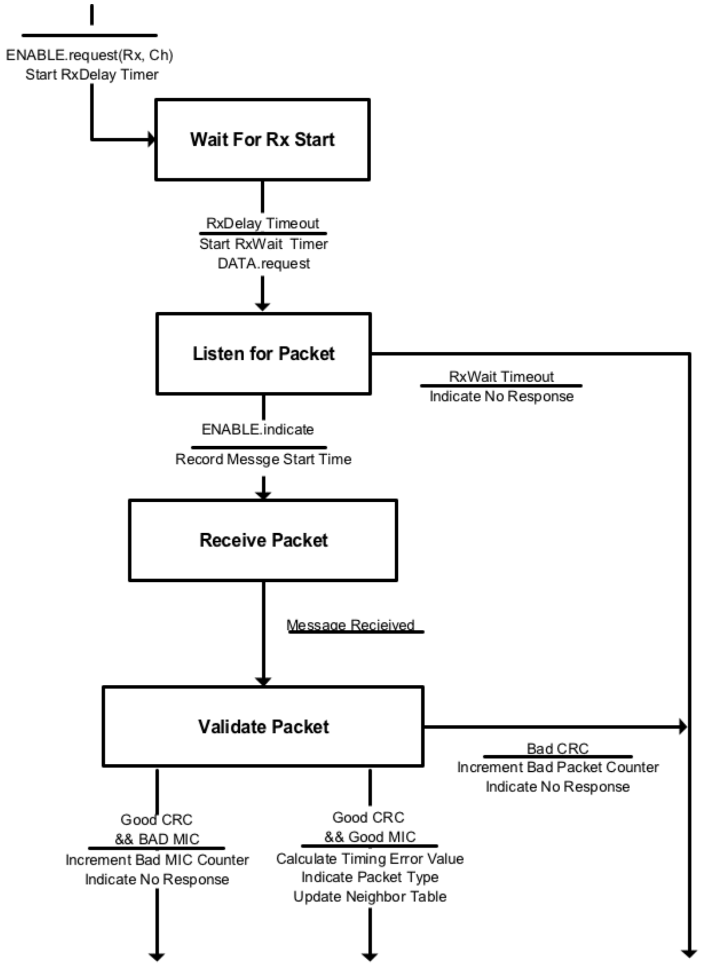


Figure 18. Receive State Machine

Subsection 10.1/Table 12

In Table 12 In Table 1 modify the TsRxAckDelay entry as follows:

TsRxAckDelay	End of message to when transceiver must be listening for ACK.	800 µs ±100 µs

This page intentionally left blank.

Table of Contents

Table of Figures

List of Tables

Preface

This preface is included for informational purposes only.

Adoption of HART and the sale of HART-enabled equipment continue to grow. There are many millions of devices installed and HART is favored by plant personnel due to its simplicity, low cost, ease of use, and high value.

Starting in late 2004, the HCF began evaluating wireless technology and developing a wireless alternative for the HART Protocol. This specification is a result of that development effort. WirelessHART provides technology for host application wireless access to existing HART-enabled field devices and supports the deployment of battery operated, wireless-only HART-enabled field devices.

WirelessHART establishes a wireless communication standard for process applications. WirelessHART further extends the application of HART Communications and the benefits it provides to industry by enhancing the HART Technology to support wireless process automation applications while meeting the following goals:

- Preserve and enhance industry's existing investment in HART Technology
- Leverage established technologies, standards, and practices to rapidly develop backward-compatible WirelessHART Standards.
- Maximize coexistence by ensuring reliable WirelessHART communications while minimizing interference with other wireless technologies.

Since the technology is fundamentally HART, existing, previously installed host applications can, without modification, access wireless-enabled HART field devices and new wireless-only HART field devices.

The following people actively served as members or made significant contributions to the creation of this document:

Gareth Johnston	ABB
Eric Rotvold	Rosemount Division, Emerson Process Management
Kelly Orth	Rosemount Division, Emerson Process Management
Mark Nixon	Emerson Process Management
Niels Aakvaag	ABB
Paula Doyle	ABB
Rick Enns	Dust Networks
Robin Pramanik	Siemens AG
Tomas Lennvall	ABB
Wally Pratt	HART Communication Foundation
Yuri Zats	Dust Networks

The Foundation and it members recognize the outstanding efforts of these people and gratefully thank their companies for supporting the development of this specification

Introduction

WirelessHART is an optional HART Physical Layer that provides a low cost, relatively low speed (e.g., compared to IEEE 802.11g) wireless connection to HART-enabled devices. The principle objectives of WirelessHART include:

- Compatibility with existing HART Application Layer;
- Leverage existing host applications and the large installed base;
- Must be HART-like: simple, reliable, easy-to-use;
- Supply end-users with new capabilities; and
- Provide more flexibility for installing and operating process automation equipment.

Furthermore, WirelessHART must be very interoperable and allow compliant devices from different manufacturers to be mixed to create an integrated system. More specifically, HART has always had a strict definition of interoperability:

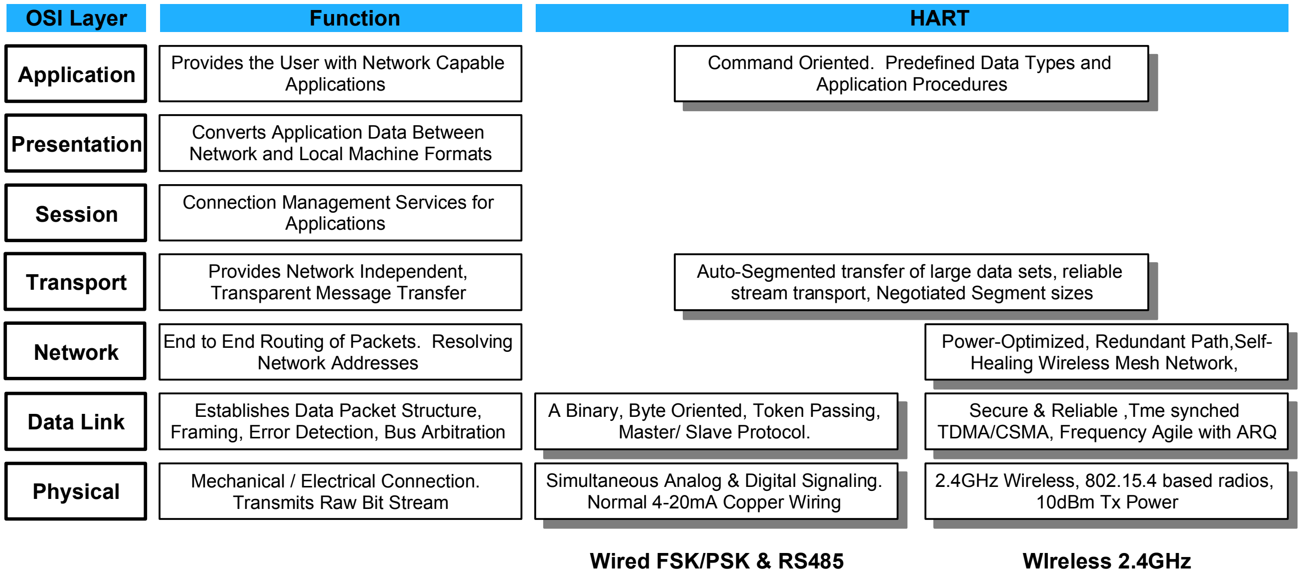
> **Interoperability** is the ability for like devices from different manufacturers to work together in a system and be substituted one for another without loss of functionality at the host system level.

WirelessHART targets unit-level process operations and supports monitoring and control applications like:

- Equipment and process monitoring;
- Asset management;
- Diagnostics/ predictive maintenance;
- Non-critical control; and
- Nomadic, "wireless worker" applications

WirelessHART is a secure, wireless mesh networking technology operating in the 2.4GHz ISM radio band. WirelessHART specifies the use of IEEE STD 802.15.4-2006 compatible 2.4GHz DSSS transceivers and uses channel hopping on a transaction by transaction basis. WirelessHART communication is arbitrated using TDMA to schedule link activity. All communication is performed within a designated slot. One or more source and one or more destination devices may be scheduled to communicate in a given slot. The slot may be dedicated to communication from a single source device or a slot may support shared communication between multiple devices. The message being propagated by the source device may be to a specific device or broadcast to all of the destination devices assigned to the slot.

HART is loosely organized around the ISO/OSI 7-layer model for communications protocols (see Figure 1). With the introduction of wireless technology to HART, two Data-Link Layers are supported: the token-passing and Time Division Multiple Access (TDMA). Both support the common HART Application Layer. In addition, since WirelessHART allows deployment of mesh topologies, a significant network layer is now specified (see the *Network Management Specification*).



OSI Layer	Function	HART	
Application	Provides the User with Network Capable Applications	Command Oriented. Predefined Data Types and Application Procedures	
Presentation	Converts Application Data Between Network and Local Machine Formats		
Session	Connection Management Services for Applications		
Transport	Provides Network Independent, Transparent Message Transfer	Auto-Segmented transfer of large data sets, reliable stream transport, Negotiated Segment sizes	
Network	End to End Routing of Packets. Resolving Network Addresses		Power-Optimized, Redundant Path,Self-Healing Wireless Mesh Network,
Data Link	Establishes Data Packet Structure, Framing, Error Detection, Bus Arbitration	A Binary, Byte Oriented, Token Passing, Master/ Slave Protocol.	Secure & Reliable ,Tme synched TDMA/CSMA, Frequency Agile with ARQ
Physical	Mechanical / Electrical Connection. Transmits Raw Bit Stream	Simultaneous Analog & Digital Signaling. Normal 4-20mA Copper Wiring	2.4GHz Wireless, 802.15.4 based radios, 10dBm Tx Power
		Wired FSK/PSK & RS485	**WIreless 2.4GHz**

Figure 1. OSI 7-Layer Model

The WirelessHART Architecture is designed to be an easy to use, reliable and inexpensive wireless mesh sensor network protocol. There are three principle elements in a WirelessHART network:

- WirelessHART field devices that are connected to the Process or Plant Equipment.
- WirelessHART gateways that enable communication between Host Applications and WirelessHART field devices in the WirelessHART network. One or more WirelessHART gateways must be connected to a given WirelessHART network.
- A WirelessHART network manager that is responsible for configuration of the network, scheduling communication between WirelessHART devices (i.e., configuring superframes), management of the routing tables and monitoring and reporting the health of the WirelessHART network. While redundant network managers are supported, there must be only one active network manager per WirelessHART network.

First and foremost, WirelessHART devices are HART devices supporting all that users have come to expect from HART. One of the core strengths of HART is its rigorous interoperability requirements. All WirelessHART equipment supports core mandatory capabilities that allow equivalent device types to be exchanged without compromising system operation. To this end the majority of WirelessHART requirements are mandatory and must be supported.

Furthermore, WirelessHART is backward compatible to HART core technology such as the Device Description Language. All HART devices (e.g., network managers, gateways, field devices, etc) shall be describable using DDL. This ensures that end users immediately have the tools to begin utilizing WirelessHART. This is critical to the rapid acceptance of WirelessHART in the market.

1. SCOPE

This specification defines the HART TDMA Data-Link Layer. This specification is normally used for mesh network communications via an IEEE STD 802.15.4-2006 Physical Layer. The Data-Link Layer is responsible for the secure, reliable, error free communication of data between HART compatible devices. In other words, this document specifies the rules used by HART products to wirelessly communicate HART digital information. Figure 2 shows the scope of this specification. This document includes:

- The services provided by the Data-Link Layer to the Network Layer. These services constitute a "black box" model of the Data-Link Layer requirements. These services are specified with the assistance of Time Sequence Diagrams.
- Logical Link Control (LLC) requirements including the format of HART frames, the structure of HART device addresses; the security services used for message integrity and the error detection coding to be used.

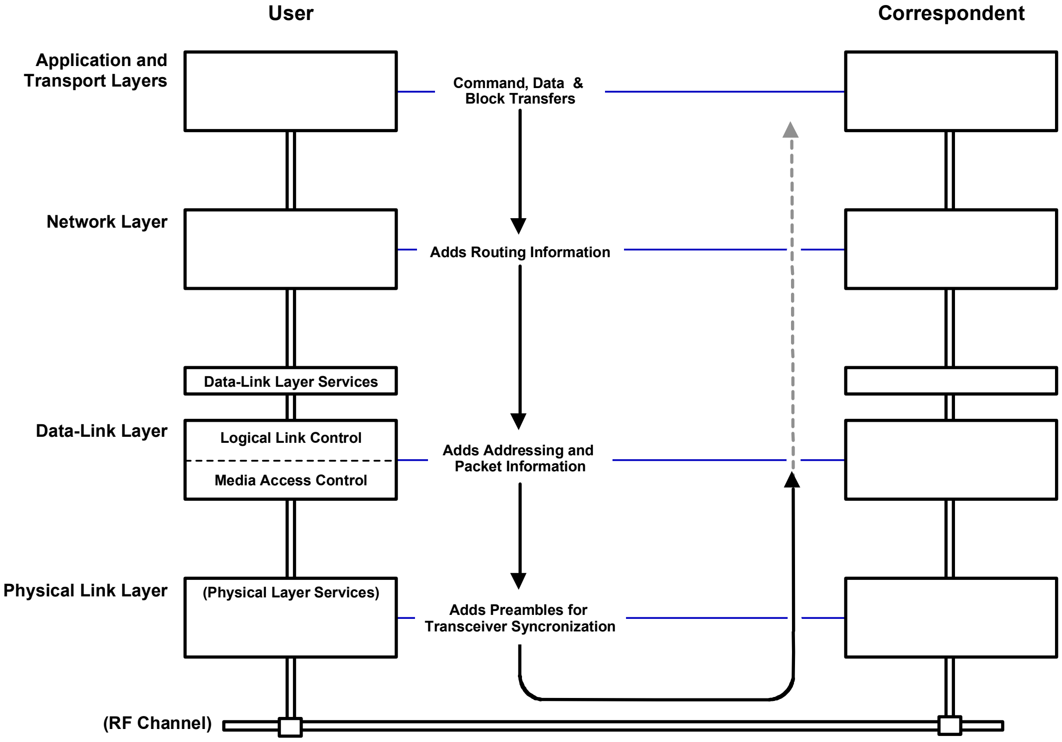

User
Correspondent
Application and Transport Layers
Command, Data & Block Transfers
Network Layer
Adds Routing Information
Data-Link Layer Services
Data-Link Layer
Logical Link Control
Media Access Control
Adds Addressing and Packet Information
Physical Link Layer
(Physical Layer Services)
Adds Preambles for Transceiver Syncronization
(RF Channel)

Figure 2. Data-Link Layer Scope

- Medium Access Control (MAC) rules ensuring that transmissions by devices occur in an orderly fashion. In other words, the MAC specifies when a device is allowed to transmit a message. MAC specifications themselves are formulated in terms of state transition diagrams, which permit an unambiguous description of the action of the MAC sub-layer.
- The actual timing values required for proper operation of the MAC sub-layer. These timing values directly correspond to Physical Layer performance characteristics (e.g., Clear Channel Assessment time, Tx/Rx turnaround time).

The segregation of requirements into these four categories is intended as a frame of reference rather than as a description of an actual implementation. While actual implementations may vary, all requirements in this specification are mandatory.

Unless specifically noted, HART data is transmitted most significant byte first (i.e., big endian).

Within this context, the Data-Link Layer as a whole has only a one hop scope. Any responsibilities to the network beyond the device's immediate neighbors are allocated to the Network Layer.

2. REFERENCES

2.1 HART Field Communications Protocol Specifications

These documents published by the HART Communication Foundation are referenced throughout this specification:

HART Field Communications Protocol Specification. HCF_SPEC-12

Token-Passing Data-Link Layer Specification. HCF_SPEC-81

Network Management Specification. HCF_SPEC-85

2.2 Related HART Documents

References to other standards, clarifying documents and applicable patents are listed in this subsection.

WirelessHART User Guide. HCF_LIT-84

Coexistence Test Plan. HCF_LIT-85

2.3 Related Documents

The following are applicable IEEE documents:

IEEE STD 802.15.4-2006. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. 2006

Diffie, W. and Hellman, M. "Privacy and Authentication: An Introduction to Cryptography". *Proceedings of the IEEE*, Vol. 67 No. 3 (March 1979). pp 397-427

In addition, the application of the IEEE Extended Unique Identifier (EUI-64™[1]) and Organizationally Unique Identifier (OUI™[2]) can be found at:

IEEE. "IEEE Registration Authority - Tutorials". IEEE Standards Association. http://standards.ieee.org/regauth/tutorials.html (accessed 1 August, 2007).

[1] EUI-64 is a trademark of The Institute of Electrical and Electronics Engineers, Inc.

[2] OUI is a trademark of The Institute of Electrical and Electronics Engineers, Inc.

The following document provides additional information about and algorithms for the 16 bit ITU-T CRC (also known as CRC16).

> Simpson, W., Editor. "*PPP in HDLC Framing*". RFC 1549. http://www.ietf.org/rfc/rfc1549.txt. IETF 1993.

The following provides general guidelines for the specification of communication protocols.

> ISO 7498-1 *Information Processing Systems — OSI Reference Model — The Basic Model*

On byte ordering

> Wikipedia contributors, "Endianness," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Endianness&oldid=105787173 (accessed 9 February, 2007).

> Cohen, Danny. "On Holy Wars And A Plea For Peace". DAV's Endian FAQ http://www.rdrop.com/~cary/html/endian_faq.html (accessed 9 February, 2007).

The following reference provides additional information about and algorithms for the CCM* Mode algorithm used in conjunction with AES-128 cipher for security.

> Dworkin, M. *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*. National Institute of Standards and Technology. Special Publication 800-38C. May 2004

The following reference describes communication specification techniques used in this document including Service Primitives (SPs) and time sequence diagrams:

> Halsall, F. *Data Communications, Computer Networks and Open Systems*. Third Edition. Addison Wesley. 1992

The following reference describes the methods for specifying state transition diagrams used in this document.

> Hatley, D., and Pirbhai, I. *Strategies for Real-Time System Specification*. Dorset House, 1987.

3. DEFINITIONS

The HART Protocol Specifications must use a common and consistent vocabulary both within a specification and across all specifications. This section incorporates (by reference) definitions from the *HART Field Communications Protocol Specification* and defines the terms unique to this specification. Vocabulary or phrases used in more than one specification must be defined in the HART Field Protocol Specification. Sometimes key definitions found in there are repeated and amplified in this section (rather then simply incorporating by reference).

Some of the following definitions are included in the *HART Field Communications Protocol Specification*. However, these definitions are critical to the understanding of this specification. As a result, they are included and their meaning amplified.

Acknowledge Explicit Data-Link response to the successful reception of a directed, non-broadcast DLPDU from a Data-Link source device. The second DLPDU of a two-DLPDU transaction.

Assailant The device generating interference.

Absolute Slot Number The count of all slots that have occurred since the network was formed and always contains the number of the current slot. The Absolute Slot Number is only incremented and must never be reset.

Byte 8-bits, sometimes called an Octet.

Channel Hopping Regular change of transmit / receive frequency to combat interference and fading.

Clear Channel Assessment Clear Channel Assessment (CCA) is used to avoid initiating a transaction while the RF channel is in use. CCA is performed by listening to the channel prior to sending the first DLPDU of a transaction.

Coexistence Coexistence is the ability of one system to perform a task in a given shared environment in which other systems have an ability to perform their tasks and may or may not be using the same set of rules (IEEE).

Data-Link Layer Layer 2 in the OSI Basic Reference Model. This layer is responsible for the error-free communication of data. The Data-Link Layer defines the message structure, error detection strategy and bus arbitration rules.

Frame A Data-Link Layer "packet" which contains the header and trailer information required by the physical medium. That is, Network Layer packets are encapsulated to become frames.

Frequency Channels The allocation of the frequency spectrum in a given frequency range.

Hop A term used to describe the data being passed from one device to another as a means to lengthen the transmit distance. Also used to denote the function of changing channels

Interoperability Interoperability is the ability for like devices from different manufacturers to work together in a system and be substituted one for another without loss of functionality at the host system level.

Link The full communication specification between adjacent nodes in the network, i.e., the communication parameters necessary to move a packet one hop.

Link Margin The difference between the power of a received signal and the sensitivity of the receiver. Typically, this determines the viability of a link. Around 10 dB of margin is desirable for a reliable link.

Logical Link Control Logical Link Control (LLC) is the higher of the two data link layer sublayers defined in the OSI Model. The LLC sublayer handles error control, flow control, framing, and addressing.

Medium Access Control A sub-layer found with the OSI Data-Link Layer (OSI Layer 2) used for arbitrating access to the communication channel.

Neighbor Adjacent nodes in the network such that the Receive Signal Level (RSL) suggests communication in at least one direction is possible.

Network Device A device with a direct Physical Layer connection to the network. Each network device (e.g., field device or gateway) has a HART Unique Address that is used in communication with the device.

Network Manager A Network Manager is responsible for configuration of the network, scheduling communication between network devices, management of the routing tables and monitoring and reporting the health of the network. There must be one and only one network manager per WirelessHART Network.

Although the network manager need not have a direct Physical Layer connection it still must have a HART Unique Address.

Packet A generic reference to the set of data communicated across a network

Physical Layer Layer 1 in the OSI Basic Reference Model. The Physical Layer is responsible for transmission of the raw bit stream and defines the physical (e.g., mechanical, electrical) connections and signaling parameters for devices.

Slot A fixed time interval that may be used for communication between neighbors.

Superframe A collection of slots repeating at a constant rate. Each slot may have several links associated with it.

Time Sequence Diagram A diagram used to illustrate the interrelationship between the Protocol services. The protocol layer of interest and the lower, intervening layers are treated as a "black box". The internal workings of these layers are not shown on this diagram. The time sequence diagram shows the interactions between the service primitives over time.

Sometimes referred to as a Message Sequence Diagram.

Transaction A complete, atomic cycle of Data-Link activity. A transaction consists of (a) a single DLPDU transmission from a source device, or (b) two DLPDUs: one from the Data-Link source followed by an second, link-level acknowledgement DLPDU from the destination.

4. SYMBOLS/ABBREVIATIONS

802.15.4™[3]	IEEE STD 802.15.4-2006 in general. When referring to the Physical Layer it refers to the 2.4GHz DSSS Physical Layer employing OQPSK modulation.
ACK	See Acknowledge
ASN	See **A**bsolute **S**lot **N**umber
CCA	**C**lear **C**hannel **A**ssessment
dBi	dBi is used to express the gain of an antenna in decibels. The terminal letter 'I' indicates that the gain is relative to an isotropic antenna.
dBm	dBm is an abbreviation for the power ratio in decibels (dB) of the measured power, referenced to one milliwatt (1 mW). 0 dBm =1 mW; 10 dBm= 10 mW; 20 dBm= 100 mW; 30 dBm= 1 W
DLL	See **D**ata-**L**ink **L**ayer
DLPDU	**D**ata-**L**ink **P**rotocol **D**ata **U**nit (i.e., a Data-Link Layer packet)
DSSS	**D**irect **S**equence **S**pread **S**pectrum
EIRP	**E**ffective **I**sotropic **R**adiated **P**ower
EUI-64	**E**xtended **U**nique **I**dentifier (64 bits long)
LSB	**L**east **S**ignificant **B**yte. The LSB is always the last byte transmitted over a HART data link
MAC	See **M**edium **A**ccess **C**ontrol.
MSB	**M**ost **S**ignificant **B**yte. The MSB is always the first byte transmitted over a HART data link.
OQPSK	**O**ffset **Q**uadrature **P**hase **S**hift **K**eying
OUI	Organizationally Unique Identifier
PDU	Protocol Data Unit. The packet of information being communicated.
PhPDU	Physical Layer Protocol Data Unit (i.e., a Physical Layer packet)
PHY	See Physical Layer
RSL	**R**eceived **S**ignal **L**evel. The signal level (in dBm) at a receiver input terminal.
STX	Start of a transaction. An STX is used to convey a Network layer packet (an NPDU) from one node to an adjacent node.
TER	**T**ransaction **E**rror **R**ate

[3] 802.15.4 is a trademark of The Institute of Electrical and Electronics Engineers, Inc.

5. DATA FORMAT

In HART Protocol command specifications, service descriptions and data table requirements, the following key words are used to refer to the data formats. For more information about these formats, refer to the *Command Summary Specification*.

Bits Each individual bit in the byte has a specific meaning. Only values specified by the command may be used. Bit 0 is the least significant bit.

Date The Date consists of three 8-bit binary unsigned integers representing, respectively, the day, month, and year minus 1900. Date is transmitted day first followed by the month and year bytes.

Enum An integer enumeration with each numeric value having a specific meaning. Only values specified in the Common Tables Specification may be used.

Float An IEEE 754 single precision floating point number. The exponent is transmitted first followed by the most significant mantissa byte.

Latin-1 A string using the 8-bit ISO Latin-1 character set. Latin-1 strings are padded out with zeroes (0x00).

Packed A string consisting of 6-bit alpha-numeric characters that are a subset of the ASCII character set. This allows four characters to be packed into three bytes. Packed ASCII strings are padded out with space (0x20) characters.

Signed-nn A signed integer where nn indicates the number of bits in this integer. Multi-byte integers are transmitted MSB – LSB.

Time The Time consists of a unsigned 32-bit binary integer with the least significant bit representing 1/32 of a millisecond (i.e., 0.03125 milliseconds).

Unsigned-nn An unsigned integer where nn indicates the number of bits in this integer. Multi-byte integers are transmitted MSB – LSB.

6. OVERVIEW

6.1 TDMA Basics

WirelessHART uses Time Division Multiple Access (TDMA) and channel hopping to control access to the network. TDMA is a widely used Medium Access Control (MAC) technique that provides collision free, deterministic communications. TDMA uses time slots where communications between devices occur. A series of time slots form a TDMA superframe. All devices must support multiple superframes, starting with superframe zero (0). At least one Superframe is always enabled while additional superframes can be enabled or disabled (see the *Network Management Specification* for more information). Slot sizes and the superframe length (in number of slots) are fixed and form a network cycle with a fixed repetition rate. Superframes are repeated continuously. Figure 3 illustrates the basics of TDMA: its slots and the superframe.

Note: While a superframe is fixed while it is active, its length can be modified when inactive and different superframes may have lengths that differ from each other.

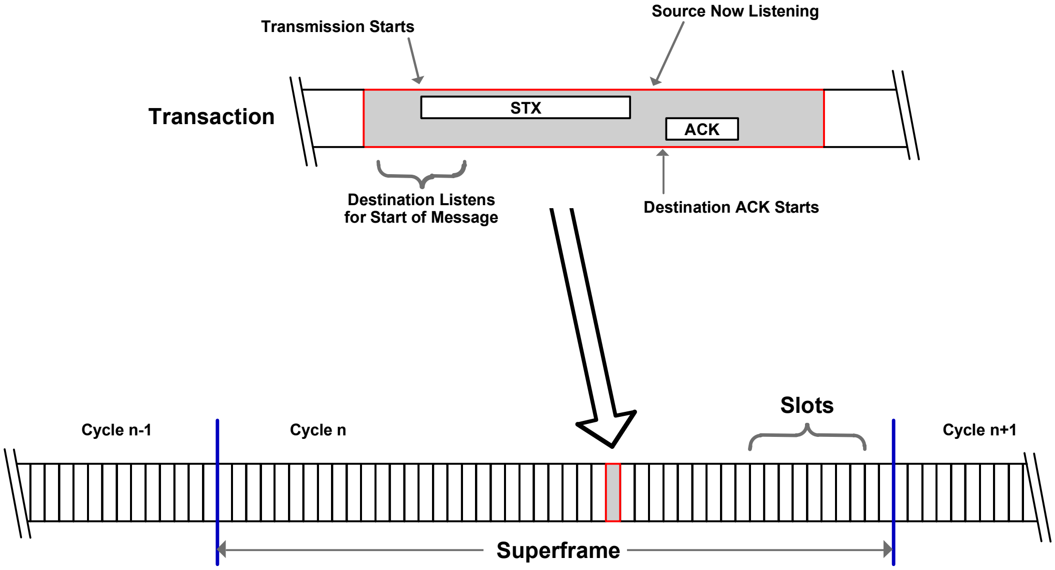


Figure 3. A TDMA Slot and Superframe

Typically, two devices are assigned to a given slot. One is designated as the source and the other, the destination. A communication transaction within a slot supports the transmission of a Data-Link Protocol Data Unit (DLPDU) from a source followed immediately by the transmission of an acknowledgement (ACK) DLPDU by the addressed device. The addressed device's response DLPDU may contain a "Success" response code indicating the initial DLPDU was successfully received and handled, or an error Response Code. An error Response Code indicates that the initial DLPDU was successfully received, but that further processing failed, e.g., there are no buffers available in the receiving device. See the *Command Response Code Specification* for more information.

Note: A broadcast message (i.e., the Data-Link destination address is the broadcast address) is never acknowledged. In this case, multiple receivers are assigned to the same slot.

For successful and efficient TDMA communications, synchronization of clocks between devices in the network is critical. Consequently, tolerances on time keeping and time synchronization mechanisms are specified to ensure network-wide device clock synchronization. It is imperative that devices know when the start of a slot occurs.

Within the slot, transmission of the source message starts at a specified time after the beginning of a slot. This short time delay allows the source and destination to set their frequency channel and allows the receiver to begin listening on the specified channel. Since there is a tolerance on clocks, the receiver must start to listen before the ideal transmission start time and continue listening after that ideal time. Once the transmission is complete, the communication direction is reversed and the destination device indicates, by transmitting an ACK, whether it received the source device's DLPDU successfully or with a specific class of detected errors. (Non-response implies either non-reception or reception with errors outside of these classes.)

To enhance reliability, channel hopping is combined with TDMA. Channel hopping provides frequency diversity, which can avoid interferers and reduce multi-path fading effects. TDMA enables efficient, low-power and reliable channel hopping communication because the synchronization of the slot and channel used by the communicating devices allows them to rendezvous in time and frequency, thus promoting successful communications (see Figure 4).

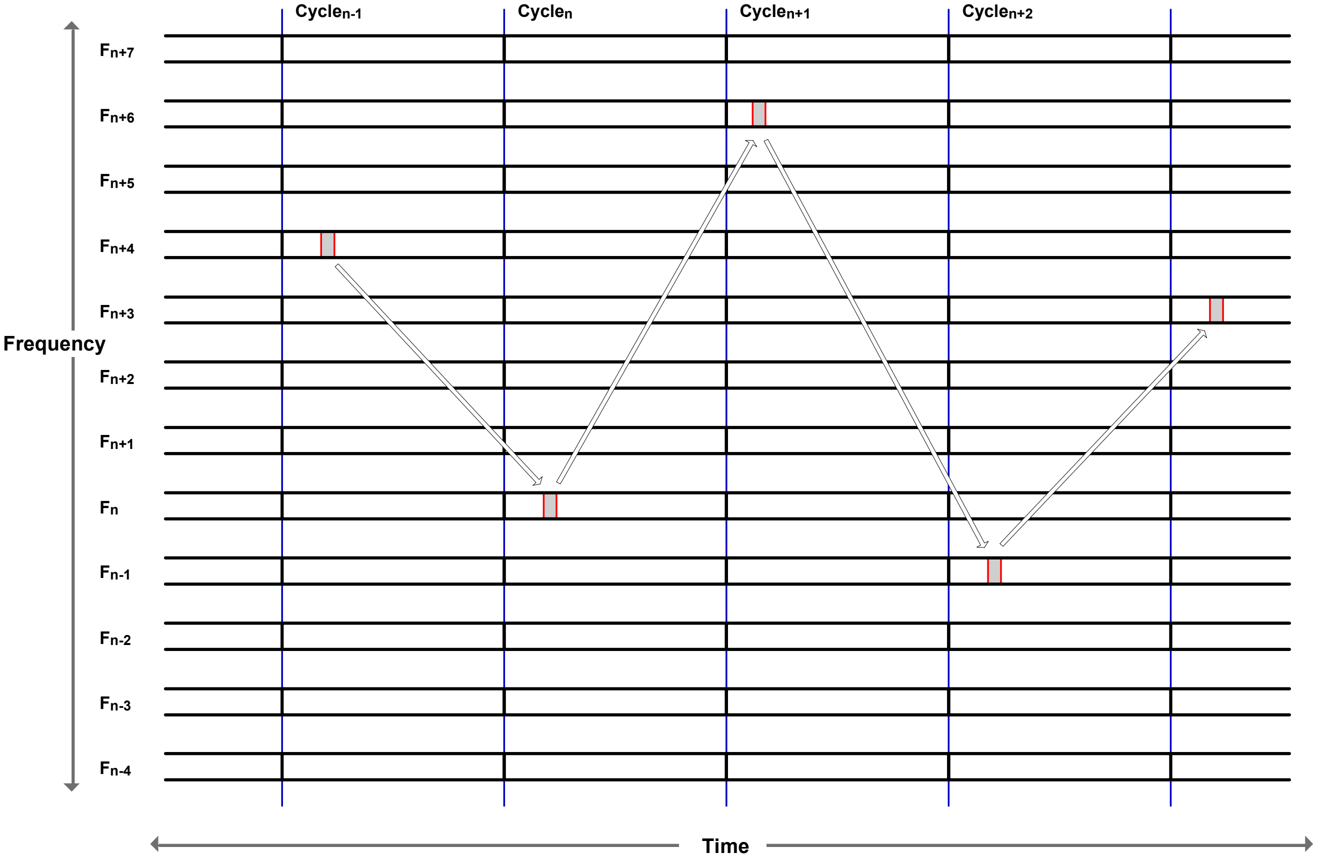


Figure 4. Channel Hopping.

Communicating devices are assigned to a superframe, slot, and channel offset. This forms a communications link between communicating devices. All devices must support multiple links. The number of possible links is, typically, equal to the number of channels utilized by a network times the number of slots in the superframe. For example, using 15 channels and 9000 slots per superframe results in 135,000 possible links.

Channel hopping provides channel diversity, so each slot may be used on multiple channels at the same time by different nodes. This can be achieved by creating links on the same slot, but with different channel offsets. Each device shall maintain a list of channels in use and the specification (e.g., the frequency) for that channel. All devices in a network shall have identical channel lists.

Assignment of links and the devices in a link is the responsibility of the Network Manager (see the *Network Management Specification*).

Channel blacklisting is the WirelessHART protocol feature that allows the network administrator to restrict the channel hopping of Network Devices network-wide to selected channels in the RF band. For example, network administrators can blacklist channels in order to protect a wireless service that uses a fixed portion of the RF band that would otherwise be shared by the WirelessHART Network. In practice, WirelessHART communication (like WiFi, Bluetooth, and other wireless communication) is very random and uses a tiny amount of the total bandwidth. Consequently, blacklisting seldom provides tangible benefits.

6.2 Mesh Networking

WirelessHART is a mesh communication protocol that simplifies installation of wireless field devices, allowing the end user to tailor the installation to the specific application requirement. WirelessHART compatible devices can be deployed in a star topology (i.e., all devices are one hop to the gateway) to support a high performance application, a multi-hop overly-connected mesh topology for a less demanding (e.g., monitoring) application, or any topology in between. In fact, WirelessHART technology is flexible enough that a variety of applications (both high and low performance) can operate in the same network.

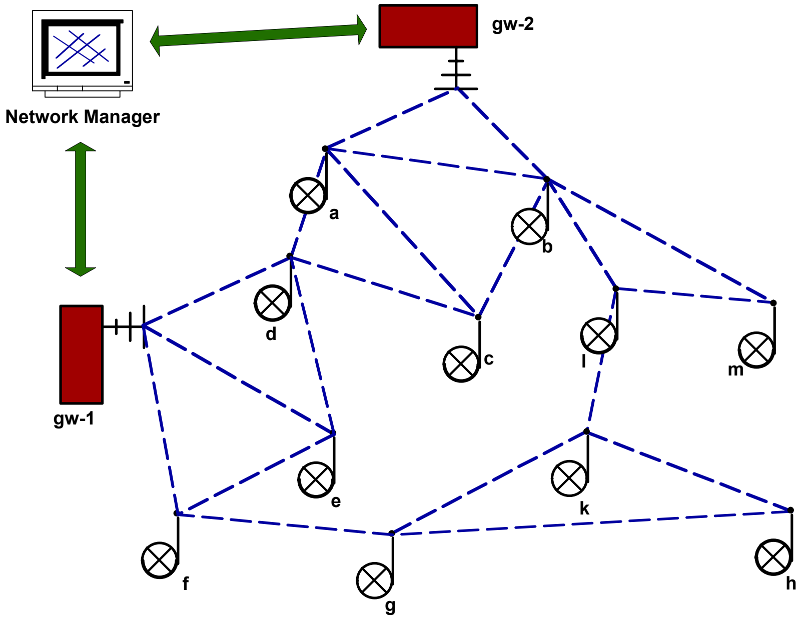


Figure 5. Mesh Network.

All network devices must be able to source and sink packets and be capable of routing packets on behalf of other devices in the network. Each time a packet moves across a link it is called a hop. The routing of packets from their initial source to their final destination may take several hops. The actual routing of packets is the responsibility of the Network Layer (see the *Network Management Specification*).

Each device shall maintain a list of links. The device cycles through links, in slot time order, servicing them as needed. Every link designating the device as the receiver must be serviced. Within that link a transmission may or may not occur. When a packet is received it is posted to the Network Layer. If the device is the packet's final destination it will be consumed. Otherwise, the Network Layer updates routing information for the next hop (if necessary), makes any required changes in packet addressing, and passes the updated packet back to the Data-Link Layer for transmittal.

The Data-Link Layer maintains a list of DLPDUs awaiting transmittal. Links designating the device as the source are only serviced when there is a DLPDU pending for the link's destination. After successful transmittal (e.g., an ACK is received) to another device, the DLPDU is discarded.

6.3 Network Maintenance

Time and channel are the first two dimensions of a WirelessHART network. The third dimension is space (distance). WirelessHART devices are installed at various locations about a plant and, consequently, a given device has a set of other devices within communications range (i.e., in its neighborhood). Since the RF environment is subject to change, a device must maintain its neighbor list. Maintenance activities include discovering potential neighbors, gathering statistics about the communication channel to each neighbor, and maintaining time synchronization with neighbors.

Two special DLPDUs, Advertise DLPDUs and Keep-Alive DLPDUs, assist in building and maintaining the device's neighbor list. The network manager can schedule transmissions of Advertise DLPDUs. These DLPDUs contain sufficient information to allow new neighbors to be discovered or to allow a newly installed device to request admission to the network. If accepted, the new device may become a neighbor of the advertising device. Further information about network joining can be found in the *Network Management Specification*.

Every successful communication with a neighbor confirms the neighbor’s presence and allows the quality of the communication link to be assessed. Since communications only occurs when the device has a DLPDU for the neighbor, there may be long intervals during which the link is not exercised. Keep-Alive DLPDUs are used to probe quiescent links and to maintain time synchronization.

6.4 Time Keeping

Time synchronization across the network is essential to TDMA communications. No matter the choice of device hardware time source (e.g., crystals, ceramic resonators etc.), some skew between devices (e.g., due to temperature or voltage variations or ageing) is inevitable. Consequently, WirelessHART has several mechanisms to promote network-wide time synchronization.

When the destination device receives a DLPDU, its time of arrival is noted. Using this information the destination calculates the difference from the ideal time at which it believes the communication should have occurred. This delta-t (Δt) must be communicated in every ACK reply DLPDU sent to the source device. Thus, every acknowledged transaction measures the alignment of network time between the devices.

Within the neighbor list, selected neighbors, specified by the Network Manger, are used as time synchronization sources. When a DLPDU from a time synch neighbor is received, the network time of the receiving device should be adjusted. Time synchronization is based either on the DLPDU arrival time or on the delta-t in the ACK, depending on which device initiated the transaction.

In addition, device designers must understand the time drift characteristics of their products. Should the device's time source drift, the device must transmit Keep-Alive DLPDUs, as needed, to its time-synch neighbors to ensure that time synchronization is maintained. Devices shall not require a Keep-Alive more often then once per 30 seconds while temperature is varying 2º C per minute or less. Furthermore, device designs must tolerate one retry in case of packet loss (i.e., a 10 second safety margin). This corresponds to approximately a compensated clock accuracy of 10ppm or better.

> Note: Keep-Alive DLPDUs are also used for neighbor discovery and to confirm the viability of quiescent links.

7. DATA-LINK LAYER SERVICES

This section specifies the operation of the TDMA Data-Link layer from a "black box" point of view. This section specifies the Service Primitives (SPs) supplied by the Physical Layer to the Data-Link Layer, which the Data-Link Layer in turn can expose to upper protocol layers (chiefly the Network Layer). In addition to specifying the individual SPs, time sequence diagrams (see [Halsall]) are included to indicate the order in which the SPs should be used and the order of event occurrence at the protocol layer boundaries. See *Token-Passing Data-Link Layer Specification* for more information on the service specification methodology.

The Services described in this section are used to obtain:

- A reliable "at least once" transaction service between peer entities. The service is not designed to provide duplicate detection.
- Management services for Data-Link Layer configuration.

All SPs described here must be supported by the device unless otherwise stated. The mapping of these SPs into an implementation is entirely a local matter and is in no way restricted by this specification.

In the definition of the SPs, parameters are defined. Some parameters are optional and may not be present in all invocations of the SP. Optional parameters are distinguished by enclosing them within square brackets ("[","]") in the SP definitions.

7.1 Message SPs

Message SPs provide services supporting the basic transfer of data between devices. There is only one required SP (`TRANSMIT`) and one optional SP (`RECEIVE`). The `TRANSMIT` SP is responsible for the message exchange. The Data-Link Layer supports automatic retransmission (i.e., retries) to ensure error-free data exchange. The time sequence diagram for these SPs is shown in Figure 6.

The transmit sequences illustrate the message traffic across the link between devices. In Figure 6 Sequence 1 shows a simple, error-free transaction. In this sequence the `TRANSMIT.request` inserts the message into the Data-Link Layer's transmit queue. Sometime later, when a slot supporting communication to the destination address occurs, the message is transmitted. When a non-broadcast message is received and validated, the correspondent Data-Link Layer transmits an ACK and generates a `TRANSMIT.indicate`. Upon reception of the ACK, or immediately when the message is to the broadcast address, the source Data-Link Layer generates a `TRANSMIT.confirm` to the requesting Network layer.

The Data-Link Layer must also allow multiple messages to be queued. Sequence 2 illustrates this requirement. Since messages in the Data-Link Layer's queue may not be posted in the same order as slots occur, the messages may not be delivered in the same order as they are enqueued. In this sequence 3 messages a queued for (possibly) 3 different Data-Link destinations. Both unicast/directed (acknowledged), and multicast/broadcast (un-acknowledged transactions are supported. Sequence 3 illustrates a multicast transaction.

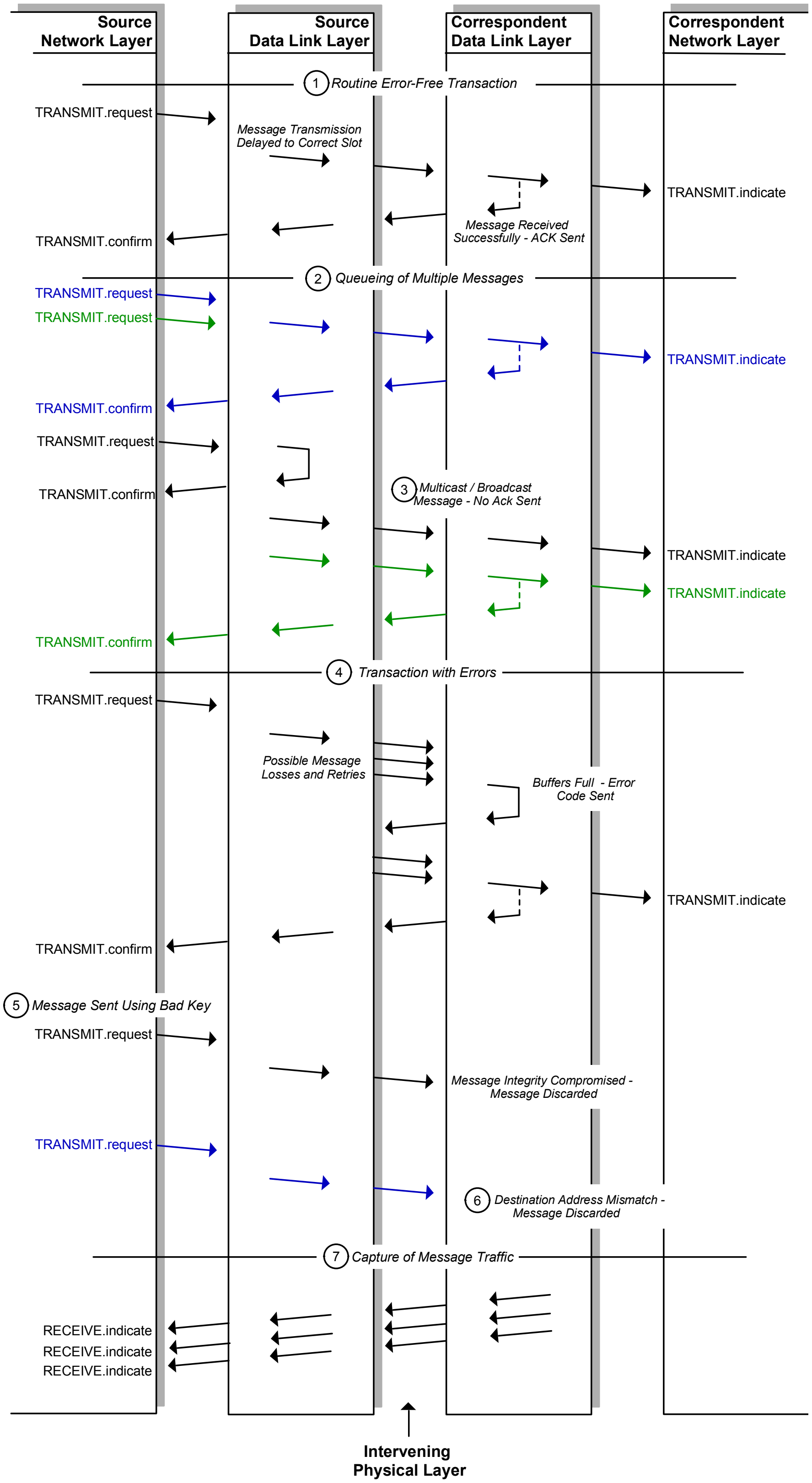

Source Network Layer
Source Data Link Layer
Correspondent Data Link Layer
Correspondent Network Layer
1 Routine Error-Free Transaction
TRANSMIT.request
Message Transmission Delayed to Correct Slot
TRANSMIT.indicate
Message Received Successfully - ACK Sent
TRANSMIT.confirm
2 Queueing of Multiple Messages
TRANSMIT.request
TRANSMIT.request
TRANSMIT.indicate
TRANSMIT.confirm
TRANSMIT.request
TRANSMIT.confirm
3 Multicast / Broadcast Message - No Ack Sent
TRANSMIT.indicate
TRANSMIT.indicate
TRANSMIT.confirm
4 Transaction with Errors
TRANSMIT.request
Possible Message Losses and Retries
Buffers Full - Error Code Sent
TRANSMIT.indicate
TRANSMIT.confirm
5 Message Sent Using Bad Key
TRANSMIT.request
Message Integrity Compromised - Message Discarded
TRANSMIT.request
6 Destination Address Mismatch - Message Discarded
7 Capture of Message Traffic
RECEIVE.indicate
RECEIVE.indicate
RECEIVE.indicate
Intervening Physical Layer

Figure 6. Message Service Sequences

The Protocol also supports automatic retries to ensure accurate data exchange (see Section 9). A transaction including retries is shown in Sequence 4 of Figure 6. As shown, the source Data-Link Layer resends the message until an acknowledge (ACK) with a Response Code of "Success" is received (or until a limit on maximum retries is reached). An individual attempt may result in no received response from the correspondent, or in receipt of an ACK with a Response Code indicating an error. If packet was not propagated then the packet will be rescheduled for a later attempt. If the packet resides in the device for an extended interval (i.e., longer than the specified packet timeout) the packet shall be discarded.

A correspondent Data-Link Layer may also receive messages with either an invalid message integrity code (see Sequence 5) or a destination address that does not match the correspondent's (see Sequence 6). These messages are normally discarded and, in all cases, must not be answered.

Sequence 7 shows the optional receive-only SP. This is available in a Data-Link Layer that support a promiscuous operating mode where communications from other devices are captured.

7.1.1 Transmit SPs

TRANSMIT.request (handle, payload, priority, timeout, graph) //graph routed
TRANSMIT.request (handle, payload, priority, timeout, sframe, bcast) //broadcast
TRANSMIT.request (handle, payload, priority, timeout, shortDestAddress) // addressed
TRANSMIT.request (handle, payload, priority, timeout, longDestAddress) // addressed
This SP is used by a device's Network Layer to transfer data to another device. The Data-Link will generate the DLPDU and initiate the transmission once a slot to the desired destination is available. All devices must be capable of queuing multiple requests and transmitting each packet on the appropriate link. The parameters to this overloaded SP include:

- ***handle*** - The handle is supported for the convenience of the client layer. The Data-Link returns this value in the corresponding TRANSMIT.confirm.

Note: This is included for clarity and is an API Artifact that is not an essential service element.

- ***payload*** - The NPDU to be propagated to the destination device.
- ***priority*** - The packet priority as determined by the contents of the payload, from the set: {**management, process-data, normal, alarm**}. See Subsection 8.3 for more information.
- ***timeout*** - The maximum time to attempt packet transmission. The Network Layer should set this based on the ASN Snippet (see the Network Management Specification).
- ***graph*** - This parameter is only present if graph routing is to be employed. When employed, the graph indicates the neighbors that may be used as the destination for the next hop.
- ***sframe*** - This parameter is only present if broadcasting a message. *sframe* indicates the superframe whose broadcast links can be used to be used to forward the packet.
- ***bcast*** - This flag indicates the NPDU must be broadcast on the indicated Superframe.

- ***shortDestAddress*** - This parameter indicates the Nickname of the destination device that must be used for the next hop
- ***longDestAddress*** - This parameter indicates the Unique ID of the destination device that must be used for the next hop.

This SP is overloaded. When an explicit address is included the Data-Link must forward the packet to that destination. Other wise the Data-Link chooses the links based on the graph routing or Superframe ID.

TRANSMIT.confirm (handle, localStatus)
This SP communicates the result of a previously issued TRANSMIT.request. The Status indicates success or failure. The handle can be used to identify the corresponding TRANSMIT.request and shall be identical to those of the TRANSMIT.request.

TRANSMIT.indicate (localStatus, priority, sourceAddress, payload)
This SP is invoked by the Data-Link Layer to notify the Network layer of a successfully received payload addressed to the device. localStatus indicates the Data-Link key used to authenticate the DLPDU and whether the DLPDU was broadcast.

FLUSH.request (handle)
Deletes the indicated packet.

FLUSH.confirm (handle, localStatus)
Indicates whether and when the packet was deleted.

7.1.2 Network Event SPs

DISCONNECT.indicate (localStatus, sourceAddress)
Notice that another device is disconnecting from the network. In other words, a DLPDU has been received from a device indicating it is leaving the network.

PATH_FAILURE.indicate (localStatus, sourceAddress)
Notice that the path to another device with which this device is connected has failed. In other words, (unexpectedly) communications with the indicated device have timed out and the device no longer seems to be available.

ADVERTISE.indicate (localStatus, AdvertisePayload)
The SP is generated upon reception of an Advertise DLPDU (see Subsection 8.2.4). Upon receiving an Advertise packet, the device that’s trying to join a network shall synchronize to the network using the Absolute Slot Number (ASN) in the packet, and posts this SP to the Network Layer.

NEIGHBOR.indicate (localStatus, sourceAddress, packetRSL)
The NEIGHBOR.indicate SP shall be generated whenever a device receives a packet from a device not listed in Neighbor Table.

7.1.3 Receive SPs

This SP is only used when the device is in promiscuous mode and, thus, forwarding all packets to the client layer.

RECEIVE.indicate (localStatus, packetRSL, payloadDLPDU)

This optional SP indicates that a frame, not addressed to this device, has been received. The local status byte carries the status of the communication as received by this device. The reception of communications from other devices often provides useful diagnostics. Sometimes this is called a "promiscuous operating mode" and can be used for network troubleshooting.

7.2 Management SPs

Management SPs support both configuration of the Data-Link Layer and access to Data-Link Layer statistics. The fundamental SP is a `LOCAL_MANAGEMENT` sequence.

> Note: None of the SPs in this section require any data to be transmitted over the communication link. Remote management of the device's Data-Link Layer configuration is possible using Application Layer messaging of standard HART commands.

These SPs allow the Data-Link Layer to be configured on power up by the device's upper layers. This also allows management of the Field Device's non-volatile and programmable non-volatile memory to be isolated from the Data-Link Layer implementation.

Management SPs may be accessed long after the Field Device has been on-line. For example, the Application Layer may receive a command from a network manager that changes the slots to be used when communicating.

LOCAL_MANAGEMENT.request (service, [data])

This SP is used to configure Data-Link Layer properties. The parameters Service and Data are defined in the table below.

LOCAL_MANAGEMENT.confirm (service, status, [data])

This SP is used to return the results of a corresponding `LOCAL_MANAGEMENT.request`. The status shall return the results of the executed request.

LOCAL_MANAGEMENT.indication (service, status, [data])

This SP is used to notify `LOCAL_MANAGEMENT` of an un-requested MAC-sublayer event report

Table 1. Local Device Management Commands

Service	Data	Description
RESET		Initializes the Data-Link Layer.
DISCONNECT		Disconnect from the network, cease communications
RE_JOIN		Disconnect from the network, rejoin the network, purging all MAC queues and clearing all MAC tables
WRITE_SUPERFRAME		Creates a new superframe.
	Unsigned-8 superframeId	
	Unsigned-16 nSlots	Length of superframe
	Boolean active	Activates (TRUE) or de-activates (FALSE) the superframe
DELETE_SUPERFRAME		Deletes an existing scheduling superframe and any associated links.
	superframeId	
ADD_LINK		Adds a new link to another device, possibly updating the neighbor and connection tables in the process.
	Unsigned-8 linkHandle	handle of created link record, if any
	Unsigned-8 superframeId	
	Unsigned-16 nodeAddress	Address of neighbor device
	Unsigned-16 slot	Slot in the superframe to use by this link
	Unsigned-8 channelOffset	Offset of logical channel relative to base channel for this slot
	linkOptions	bitmap: {Transmit=b001, Receive=b010, Shared=b100}
	linkType	One of {NORMAL, JOIN, DISCOVERY}
DELETE_LINK		Deletes an existing link, possibly updating the neighbor and connection tables in the process.
	linkHandle	
ADD_CONNECTION		Adds a new connection to a device via a specified graph
	Unsigned-8 connectionHandle	Handle for this connection
	Unsigned-16 graphId	graphId for the connection
	Unsigned-16 nodeNickname	address of device being connected via the specified graph
DELETE_CONNECTION		Deletes an existing connection
	Unsigned-8 connectionHandle	Handle of an existing connection

Service	Data	Description
READ_NETWORKID		Reads the ID of the network the device belongs to.
	Unsigned-16 NetworkID	
WRITE_NETWORKID		Writes the ID of the network the device belongs to.
	Unsigned-16 NetworkID	
WRITE_NETWORK_KEY		This command allows the Network Manager to write the network key on a Network Device. Keys should be protected from pilfering (e.g., by encryption)
	Unsigned-128 networkKey	
	Unsigned-48 SlotNumber	Execution time for command (ASN). (0 means execute immediately)
READ_TIMEOUT _PERIODS		
	Time keepAliveInterval	Interval during which a node must successfully communicate with each linked neighbor. Any DLPDU received from the neighbor resets the Keep-Alive timer for that neighbor.
	Time pathFailInterval	Interval of unsuccessful communication with a given neighbor, indicating a path failure
	Time advertiseInterval	Time period specifying the transmission of Advertise DLPDUs
	Time discoveryInterval	Time period specifying the interval bounding the random transmission of Advertise DLPDUs on Discovery links.
WRITE_TIMEOUT _PERIOD		Write the indicated time period value.

	Unsigned-8 timerCode	One of { Keep-Alive;
		Path-Failure;
		Advertise; or
		Discovery
		}

	Time timerPeriodValue	
READ_CAPACITIES		
	Unsigned-16 maxSuperframes	
	Unsigned-16 maxLinks	
	Unsigned-16 maxNeighbors	
	Unsigned-16 maxPktBuffers	
READ_PRIORITY _THRESHOLD		
	Unsigned-4 priorityThreshold	Specifies the lowest priority DLPDU to be accepted from another device.
WRITE_PRIORITY _THRESHOLD		
	Unsigned-4 priorityThreshold	
READ_JOIN_PRIORITY		Indicates what join priority the device should advertise. Lower number indicates a better choice for joining.
	Unsigned-4 joinPriority	

Service	Data	Description
WRITE_JOIN_PRIORITY		
	Unsigned-4 joinPriority	
READ_PROMISCUOUS _MODE		Indicates whether the sublayer is in “receive all” mode. TRUE indicates the sublayer accepts all PDUs received from the Physical Layer.
	Boolean promiscuousMode	
WRITE_PROMISCUOUS _MODE		
	Boolean promiscuousMode	
READ_MAX_BACK_OFF _EXPONENT		The maximum value that can be assumed for the back-off exponent used in shared slots. Valid values are { 4, 5, 6, 7 }
	Unsigned-4 MaxBackoffExponent	
WRITE_MAX_BACK _OFF_EXPONENT		
	Unsigned-4 MaxBackoffExponent	

8. LOGICAL LINK CONTROL

8.1 The DLPDU

This subsection specifies the format of the Data-Link packet (DLPDU). Each DLPDU consists of the following fields:

- A single byte set to 0x41
- A 1-byte address specifier;
- The 1-byte Sequence Number;
- The 2 byte Network ID
- Destination and Source Addresses either of which can be 2 or 8-bytes long;
- A 1-byte DLPDU Specifier
- The DLL payload
- A 4-byte keyed Message Integrity Code (MIC), and
- A 2-byte ITU-T CRC16

Figure 7 illustrates the basic PhPDU and DLPDU structure.

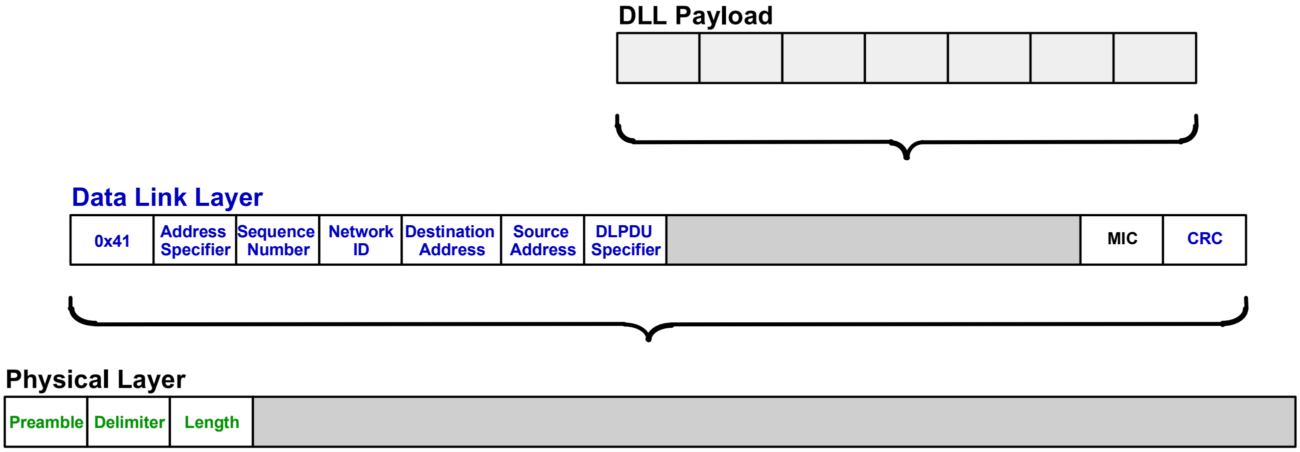


Figure 7. DLPDU Structure

8.1.1 The Sequence Number

The Sequence Number shall be set equal to the least significant byte of the Absolute Slot Number (ASN).

8.1.2 The Network ID

All networks are identified using a 2-byte Network ID. This 2 Byte value is transmitted, LSB first, in all DLPDUs. If the Network ID does not match that of the network the device is a member of then the packet is discarded. The ranges of Network ID values and their application are shown in Table 2.

> Note: Only in the header fields for WirelessHART is this byte ordering followed. Unless specifically noted, data is transmitted most significant byte first in all HART communications (i.e., big endian).

Table 2. Network ID Allocation

Range	Application
00000-32767 (0x0000-0x7FFF)	Permanent User defined networks (Critical networks)
32768-36863 (0x8000-0x8FFF)	Temporary User defined networks (Demos, trade shows, field trials, etc.)
36864-57343 (0x9000-0xDFFF)	Reserved
57344-61439 (0xE000-0xEFFF)	Manufacturing networks (non-public used by device manufacturers)
61440-65535 (0xF000-0xFFFF)	Reserved

8.1.3 Destination and Source Addresses

WirelessHART supports two types of addresses: a 2-byte "nickname" and an 8-byte IEEE EUI-64 address. The addresses contained in a DLPDU are indicated in the Address Specifier field (see Figure 8). Setting the bit 6 indicates a long 8-byte source address is contained in the DLPDU. Setting bit 2 indicates a long 8-byte destination address is contained in the DLPDU. Any combination of address lengths may be used in a DLPDU. The other bits must be set as indicated in Figure 8.

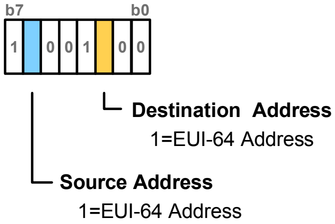


Figure 8. Address Specifier

The 2-byte nickname is assigned and managed by the Network Manager. Consequently, it is only locally unique (i.e., within the network the device belongs to). Two-byte addresses either indicate a specific network device or they may specify the broadcast address (i.e., 0xFFFF).

The EUI-64 address consists of a 3 byte "Organizationally Unique Identifier" (OUI) (assigned by IEEE) and the 5-byte Unique ID (controlled by the HART Protocol). For WirelessHART the EUI-64 shall be constructed using HCF's OUI (which is 0x001B1E) concatenated with the 40-bit HART Unique ID as shown in Figure 9. DLPDUs received with EUI-64 addresses that do not specify the HCF OUI shall be discarded.

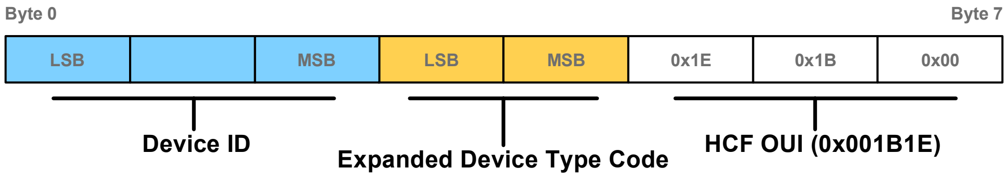


Figure 9. Construction of 8 byte EUI-64 Addresses

The Unique ID is the concatenation of the 2-byte Expanded Device Type Code and the 3-byte Device Identifier. The Expanded Device Type Code is allocated by the HCF. Further specifications regarding the use of Device Type codes can be found in the *Command Summary Specification*. Each device manufactured with the same Device Type Code must have a different Device ID

IEEE STD 802.15.4-2006 requires multi-byte fields to be transmitted LSB first (little endian) and the WirelessHART addressing is compliant. Consequently, the long address is transmitted in the DLPDU starting with the LSB of the Device ID and ending with the MSB of the HCF's OUI. The nickname is also transmitted little-endian (LSB first).

> Note: Only in the header fields for WirelessHART is this byte ordering followed. Unless specifically noted, data is transmitted most significant byte first in all HART communications (i.e., big endian).

8.1.4 DLPDU Specifier

The DLPDU Specifier is transmitted after the Network ID and addresses (see Figure 10). The most significant two bits are reserved and no device shall make any assumption regarding their possible future use. Implementations must mask off the most significant two bits. Devices built before any such future use is assigned shall set these bits to zero on transmission.

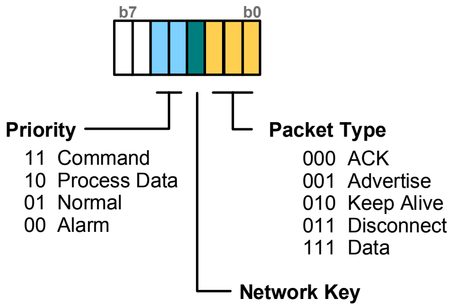


Figure 10. DLPDU Specifier

The next two bits (i.e., bits 4 and 5) indicate the priority of the message (see Subsection 8.3). Command level is the highest priority and Alarm level is the lowest priority.

Bit 3 indicates the key being used to authenticate the DLPDU. All communications within the network between authenticated device must set this bit and use the confidential Network-Key when generating the DLPDU MIC (see Subsection 8.4 for more information). This bit shall only be reset during the joining process (see the *Network Management Specification*) while the device is unauthenticated. DLPDUs passed between the unauthenticated device and its neighbor shall use the well known key from this Specification to generate the DLPDU MIC.

The least significant 3 bits of the DLPDU Specifier indicate the DLPDU type. There are five DLPDU types: Data, Keep-Alive, Advertise, Disconnect, and ACK DLPDUs.

8.1.5 DLL Payload

The DLL payload depends on the DLPDU type (defined in the DLPDU Specifier field). Data DLPDUs contain a Network Layer header and payload. Data-Link command DLPDUs have contents that depend on the type of command DLPDU. For example, acknowledgement DLPDUs have a payload that consists of a time adjustment as measured by the destination device specified in the prior DLPDU.

8.1.6 Keyed Message Integrity Code (MIC)

A keyed Message Integrity Code (MIC) is used for link-layer authentication of DLPDUs (see Subsection 8.4 for more information). Devices shall reply only to unicast, non-acknowledgement DLPDUs that have been successfully authenticated.

8.1.7 Cyclic Redundancy Check (CRC)

The Cyclic Redundancy Check (CRC) Field is based on the 16 bit ITU-T CRC polynomial (also known as a CRC16). The CRC is calculated over the entire frame using the following polynomial:

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

The CRC is usually calculated in hardware. For more information see RFC 1549 and IEEE STD 802.15.4-2006.

At receivers, the received byte stream is subjected to a similar calculation. When a receiver can predict the bytes that will contain a CRC, it may omit those bytes from the calculation and check that the received CRC matches those bytes. Alternatively, the receiver may compute a CRC over all the bytes of the message and check that the result matches an expected residual value of zero.

The CRC is used to detect bit errors and devices shall reply only to non-acknowledgement DLPDUs with a CRC matching that calculated by the receiving device, or whose residual value after processing the entire DLPDU, including received CRC bytes, matches the expected residual of zero.

8.2 DLPDU Types

The least significant 3 bits of the DLPDU Specifier indicate the type of DLPDU being communicated and the purpose of the (optional) DLL payload. There are five DLPDU types:

- Data DLPDUs contain network and device data in transit to their final destination device. The source and sink for Data DLPDUs is the Network Layer.
- Keep-Alive DLPDUs facilitate connection maintenance between neighboring devices.
- Advertise DLPDUs provide information to neighboring devices wishing to join the network.
- Disconnect DLPDUs are used to advise neighboring devices that the device is leaving the network.
- ACK DLPDUs are the immediate link level response to receipt of the source device's transmission DLPDU.

Devices receiving a packet with an unknown packet type must not acknowledge the packet and shall immediately discard it.

ACK, Advertise, Keep-Alive and Disconnect DLPDUs are generated and consumed by peer Data-Link Layers. These packets are not propagated to the Network Layer or onward through the network. In other words, these are a DLL command packets originated within the source Data-Link and consumed by a neighboring, destination peer Data-Link.

8.2.1 Data DLPDUs

Data DLPDUs contain data in transit to a final destination device. The packet payload in these DLPDUs originates from the Network Layer of this hop's source device and is passed to the destination device's Network Layer. From there these payloads are forwarded by the peer Network Layer to their final destination.

8.2.2 ACK DLPDUs

ACK DLPDUs are transmitted by a device in response to receipt of a non-broadcast, non-ACK DLPDU. All successfully received unicast, non-ACK DLPDUs must initiate the transmission of an ACK DLPDU. The ACK DLPDU shall always calculate its MIC using the same key used in the received DLPDU. See Subsection 8.4 for more information about Generating a MIC and keys.

The ACK contains a Response Code that indicates whether the receiving device has accepted the DLPDU. The destination device shall respond with "Success" (RC=0) when the packet is accepted by the device or indicates the reason why the packet was not accepted (e.g., RC=61 if the destination device is out of buffer space). Destination devices must respond with an ACK DLPDU in response to all Keep-Alive, Advertise, or Disconnect DLPDUs addressed to the device and successfully received.

To manage packet flow through the network, the network manger can raise or lower the priority level of packets that the device may accept. Devices must accept DLPDUs with a priority greater than or equal to the current priority level set by the Network Manager. In addition, space allowing, the device must accept DLPDUs whose final destination is the device itself.

In addition to the Response Code, the ACK payload includes the Time Adjustment field. The Time Adjustment is the difference between the expected time of reception of the complete start delimiter that frames a non-ACK DLPDU and the actual reception of that complete start delimiter, measured in microseconds. The Time Adjustment is a 2-byte, two's-complement integer.

The ACK payload and response codes are as follows:

ACK DLPDU Payload

Byte	Format	Description
0	Unsigned-8	Response Code
1 - 2	Signed-16	Time Adjustment in microseconds. The value of the Time Adjustment is positive (negative) if the DLPDU was received earlier (later) than expected.

ACK Response Codes

Code	Class	Description
0	Success	Success. DLPDU accepted by destination device.
61	Error	No Buffers Available. DLPDU Discarded.
62	Error	No Alarm/Event Buffers Available. DLPDU Discarded.
63	Error	Priority Too Low. DLPDU Discarded.

8.2.3 Keep-Alive DLPDUs

The Keep-Alive DLPDU is a Command DLPDU used, as needed, for network maintenance. For Keep-Alive DLPDUs, the payload field is empty. Keep-Alive DLPDUs may be used:

- For network time synchronization. Time synchronization is updated based on the Time Adjustment value returned in the corresponding ACK.
- To assess communication with a neighbor (e.g., to confirm connectivity).
- In Neighbor Discovery. When instructed the device shall send Keep-Alive DLPDUs periodically to allow the device to be detected by others.

8.2.4 Advertise DLPDUs

The Advertise DLPDU is used to invite new devices into the network. When a device wishes to join a network, it listens for these DLPDUs and then uses the information in the DLPDU to synchronize with the network and initiate the join process.

The Advertise packet includes basic network information including: ASN, the join control information, and the security levels support by the network. In addition, the channel map array is included in the Advertise DLPDU. The absolute slot number and the channel map allows the current channel offset to be identified when issuing a packet to petition the Network Manager for admission to the network. The size of the channel map array depends on the Physical Layer in use. For example, with the IEEE STD 802.15.4-2006 2450MHz Physical Layer it is two bytes long.

Once the basic network information is disclosed, the Advertise DLPDU lists all of its join links by superframe. In addition, each link is identified as either transmit or receive. The joining device shall assume all join links with a device are shared. The joining device is limited to these links until it is authenticated by the Network Manager, and has received its network and session keys. By limiting communication to only the join links the Network Manager, in effect, can quarantine the device until it is ready for the device to fully participate in the network. The format of this DLPDU is as follows:

Advertise Payload Format

Byte	Format	Description
0-4	Unsigned-40	Absolute slot number. The number of slots since the start of the network to the slot used for transmission of this DLPDU.
5	Bits	Join Control.
5.7-5.4	Enum-4	Security level supported (see Common Table 53)
5.3-5.0	Unsigned-4	Join Control - Join Priority. An unsigned integer indicating the ability of the advertising device to support another child device. The lower the value the better this advertising device is to join.
6	Unsigned-8	Number of bits of channel map array. Maximum size of the channel map array is 64 bits.
7-*n*	Bits [ ]	Channel map array. This is an array of bits starting with the least significant bit (bit 0 of byte 0) and adding bytes as necessary until all bits are accounted for. Each bit corresponds to a channel. If the bit is set the corresponding channel is in use.
n+1	Unsigned-16	Graph ID
n+3	Unsigned-8	Number of superframes.

For each Superframe

	Unsigned-8	Superframe ID
	Unsigned-16	Superframe size. The number of slots in this superframe.
	Unsigned-8	Number of Links.

For each Link

	Unsigned-16	Join slot. The specific slot within the superframe for this Link.
	Bits	Join slot channel offset.
	x.7	Reserved. Must be set to zero. No device shall make any assumption regarding their possible future use of this bit.
	x.6	When set, the link is for DLPDU transmission.
	x.5-x.0	Channel offset. The frequency channel offset for this slot. This value is used to calculate the link frequency/channel.

An Advertise packet may be sent on any transmit link that is not in use. All other Data-Link activities have a higher priority than issuing an Advertise. Advertise DLPDUs shall always calculate their MIC using the well-known key (see Subsection 8.4 for more information).

8.2.5 Disconnect DLPDUs

Disconnect DLPDUs are generated by devices leaving the network. This means the device is no longer available for communication and must be removed from the neighbor list. In addition, all links connecting the neighbor to this device shall be deleted as well. The Network Layer shall be notified when a Disconnect DLPDU is received.

For Disconnect DLPDUs, the payload is empty. Disconnect DLPDUs always calculate their MIC using the network key (see Subsection 8.4 for more information).

8.3 DLPDU Priority and Flow Control

The priority of a DLPDU is dictated by its contents. There are four priority levels:

- Command (highest priority). Any packet containing a payload with network-related diagnostics, configuration, or control information shall be classified with a priority of "Command".
- Process-Data. Any packet containing process data (e.g., Command 3 or 9) or network statistics (e.g., Command 779, 780) shall be classified as priority level "Process-Data". Only the control of the network (as indicated by the "Command" priority) is more important than delivery of measurements from process transmitters or setpoints to control devices. Process-Data priority packets must be refused from other devices when three-quarters of the device's packet buffers are occupied.
- Normal. DLPDUs not meeting the criteria for "Command", "Process-Data", or "Alarm" shall be classified as "Normal" priority. Normal priority packets must be refused from other devices when one-half the device's packet buffers are occupied.
- Alarm (lowest priority). Packets containing only alarm and event payload shall assume a priority of "Alarm". Devices shall buffer no more then one DLPDU having "Alarm" priority.

Since multiple Application Layer commands can be aggregated into a single message, the DLPDU shall assume the priority of the highest priority Application Layer command in the DLPDU.

The priority of the DLPDU is used for flow control to mitigate network congestion and ensure the Network Manager retains control of the network during a process upset or when an adverse RF event occurs. For example, the Network Manager can raise the priority threshold to reduce packet flow through the device.

In addition, the priority setting of a DLPDU has fundamental filtering effects that are applied as packets are received. Upon receiving a DLPDU, the device will use the priority of the DLPDU and the current priority threshold to determine whether the packet is accepted or discarded as follows:

- Keep-Alive, Advertise and Disconnect DLPDUs must always be received, accepted and generate an ACK with a "Success" response code.
- DLPDUs received with "Command" level priority should always be accepted and either consumed or forwarded. At least one buffer should be reserved for command packets.
- A DLPDU with "Alarm" priority shall be accepted only if the single buffer reserved for that class of DLPDU is available.
- For all other received DLPDUs, the packet priority is compared to the priority threshold level. Received packets with lower priority must be discarded. Furthermore, if the device does not have packet buffers available for that DLPDU, it must be discarded.

In summary, network management packets always propagate through the network allowing the Network Manager to keep the network operational. Alarms flow through the network is restricted ensuring alarm floods do not disrupt network operation. Since alarms are always time-stamped, no information regarding, for example, failure sequences is lost.

Finally, all other network traffic flows through the network as buffer space and bandwidth allows. Within this network traffic, process data has priority. Operation and control of the process is second only to preventing network communication disruption.

8.4 Error Detection Coding and Security

To perform error detection and to ensure network security, WirelessHART includes an unkeyed CRC and a MIC on every DLPDU. The CRC is used to detect communications errors. The CRC is calculated across the entire DLPDU using the 16-bit ITU-T algorithm (see Subsection 8.1.7).

8.4.1 MIC Calculation

A keyed MIC is used to ensure that the DLPDU originated from an approved, authenticated device. The DLPDU itself is not enciphered; rather its contents are authenticated using the four-byte MIC The MIC is generated and confirmed using CCM* mode (Counter with CBC-MAC (corrected)) in conjunction with the AES-128 block cipher to provide authentication. This cipher requires four byte-strings as parameters:

- 'a', the additional data to be authenticated but not enciphered;
- 'm', is the message to be enciphered;
- 'N', the 13-byte nonce; and
- 'K', the 128-bit AES Key.

Since the DLPDU is not enciphered, the byte-string 'm' is empty (i.e., its length is zero). The DLPDU, from the 0x41 byte through the end of the payload, is the byte-string 'a'.

DLL Keys

The key is 128-bits long (16 bytes) and, as per the CCM Mode requirements, is copied into the 'K' byte-string in most significant byte first. In other words, K[0] and K[15] are the most significant and least significant byte of the key, respectively.

There are two DLL keys: the well-known key (used in advertisements and when joining the network), and the network key (used for all other transactions). The well-known key is identical for all WirelessHART devices and has a value of **`7777 772E 6861 7274 636F 6D6D 2E6F 7267`** hexadecimal. The well-known key is used for messages passed between the joining device and devices already part of the network.

The network key is a write-only value controlled by the network manager and used for all DLL transactions except Advertise and Join DLPDUs. This key is supplied by the Network Manager to joining devices and the Network Manager may change the network key from time to time.

DLL Nonce

The 'N' byte-string must be exactly 13 bytes long and is the concatenation of the ASN and the source address.

The ASN is the count of all slots that have occurred since forming the network. It is only incremented, must never be reset. In other words, the ASN always contains the number of the current slot. The ASN is 5-bytes long and is copied (MSB to LSB) into N[0] to N[4].

The final 8-bytes of the nonce contain the source address. If the DLPDU has the EUI-64 address then it is copied (MSB to LSB) into N[5] through N[12]. In other words, 0x00, 0x1b, 0x1E are copied into N[5] through N[7], respectively. The Expanded Device Type Code and Device ID are copied (MSB to LSB) into N[8] through N[9] and N[10] through N[12], respectively.

If the 2-byte Nickname is used in the DLPDU then the nickname is copied (MSB to LSB) into N[11] through N[12]. N[5] through N[10] are set to 0x00.

8.4.2 Errors

Two errors may occur and, in both cases, result in the DLPDU being discarded and no response being generated by the destination device. The first potential error is a CRC mismatch. When the DLPDU is first received, the CRC is checked. If the CRC in the message does not match that in the DLPDU, the DLPDU is discarded.

The second potential error is an authentication failure. After confirming the CRC, the MIC is calculated and compared to the MIC in the DLPDU. If they disagree, the DLPDU is not authentic and it is discarded.

9. MEDIUM ACCESS CONTROL

The primary objectives of the Medium Access Control (MAC) sublayer are to maintain slot synchronization, identify slots that must be serviced, listen for packets being propagated from neighbors and, in turn, propagate packets received from the Network Layer. Fundamentally, the Medium Access Control (MAC) sub-layer is responsible for propagating DLPDUs across a link. To accomplish this, the device includes:

- Tables of neighbors, superframes, links, and graphs that configure the communication between the device and its neighbors (see Subsection 9.2). These tables are normally populated by the Network Manager. In addition the neighbor table is populated as neighbors are discovered.
- A link scheduler (see Subsection 9.3) that evaluates the device's tables and chooses the next slot to be serviced by listening for a packet or by sending a packet. In general the link scheduler walks the tables to identify the next slot in which to send a packet and the next slot in which to listen. The slot scheduled is the next of these two slots to occur.
- State machines that control the propagation of packets through the MAC sub-layer. MAC Operation (see Subsection 9.4) consists of schedule maintenance and service slots. MAC Operation is fundamentally event driven and responds to service primitive invocations and the start of slots needing servicing.

The number one priority of the MAC sublayer is to propagate packets enqueued in the device's buffers. Its next priority is to receive packets from neighboring devices. Both of these operations are performed one slot at a time either by sending a DLPDU or by listening for one. Successful communications depends on slot time synchronization between neighbors and the compliant timing of the transaction within the slot.

9.1 Slot Timing

All transactions occur in slots following specific timing requirements. Figure 11 shows one slot and provides an overview of transaction timing. The top of the timing diagram shows the operation of the source neighbor and the bottom shows the destination neighbor. In the figure, the destination's perception of the slot start time is slightly retarded when compared to the source's. All of the timing symbols are depicted even though they may not be applicable to every type of transaction. Table 3 defines the timing symbols.

Each slot begins by allowing a time interval to prepare the packet being conveyed for transmission. This includes formatting of the packet and calculation of the MIC and CRC. Of course these calculations are only performed if the source has a packet to propagate to the destination. The source will perform the CCA (when required) and transmit the packet. Depending on the type of transaction an ACK may be transmitted by the destination device.

When scheduled as the link's destination, the device must enter receive mode. The device must be listening for communication, starting TsRxOffset from the start of its slot, before and after the device's estimation of the ideal transmit start time. The receive window (specified by TsRxWait) allows device timing to drift while still permitting devices to communicate and resynchronize their slot timers. Sources of drift include temperature, aging, and other effects.

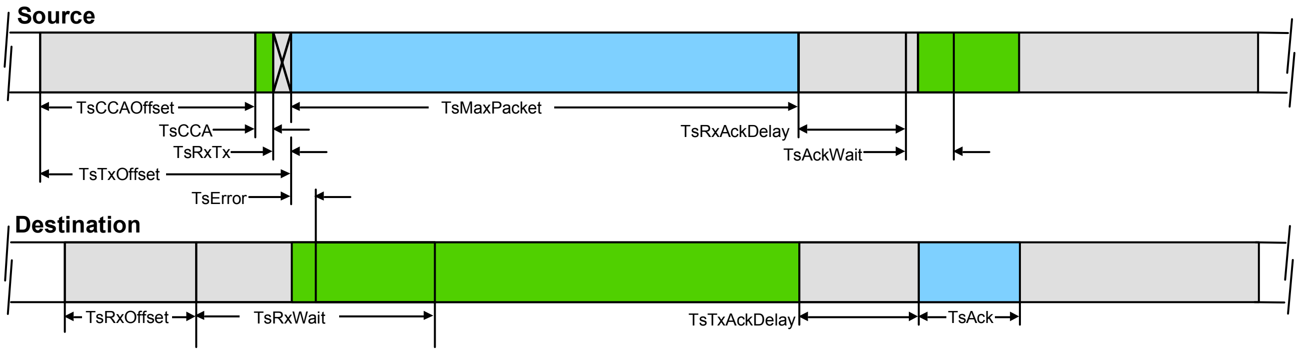


Figure 11. Slot Timing

If the destination device detects a message, it captures the time when the start of message (i.e., the end of reception of the Physical Layer Delimiter) occurs and calculates TsError as the difference between the device's ideal start time and the actual start time of the packet.

If a specific destination address is specified, the source packet will result in the destination device generating and transmitting an ACK packet. If the destination address is the broadcast address no ACK packet is generated. Finally, time is allocated at the end of the slot for processing the propagated packet and preparing for the next slot, (e.g., assessing and prioritizing the packets now queued up in each device). If one of the neighbors was the time source for the other then the end of the slot time will be aligned after successful communication.

Acknowledged Transactions

Most communications consists of the source device propagating a message by transmitting a packet and the destination device acknowledging the reception of that packet. For acknowledged communication the source and destination address in the DLPDU must contain a specific device address (i.e. not a broadcast address).

The source device must begin its transmission such that the Start of Message (SOM) occurs exactly TsTxOffset after its start of slot. SOM occurs upon completing the reception of the Physical Layer Delimiter. When performed, the CCA is performed beginning at TsCCAOffset after the start of the slot. The CCA is performed (TsCCA) and, if the channel is occupied, the transaction attempt is rescheduled for a later slot. Otherwise the transceiver is switched from receive to transmit (TsRxTx) and the packet is transmitted.

The destination device must enter receive mode and be listening for communication by TsRxOffset from its start of the slot. The destination must listen for the SOM for a duration of TsRxWait. If the destination device detects the SOM then it must receive and validate the message. Any message that cannot be validated must not be acknowledged.

Table 3. Slot Timing Symbols

Symbol	Description
TsTxOffset	Start of the slot to start of preamble transmission.
TsRxOffset	Start of the slot to when transceiver must be listening.
TsRxWait	The minimum time to wait for start of message. This correlates to the amount of drift between the neighbors that can be tolerated and communications still be maintained.
TsError	This is the difference between the actual start of message and the ideal start of message time as perceived by the receiving device. In other words, this is how much the receiving device perceives the transmitting device to be out of sync.
TsMaxPacket	The amount of time it takes to transmit the longest possible message (includes PhL preamble, delimiter, length and DLPDU
TsTxAckDelay	End of message to start of ACK. The destination device must validate the STX, and generate an ACK during this interval. Note: Broadcast messages are not acknowledged.
TsRxAckDelay	End of message to when transceiver must be listening for ACK.
TsAckWait	The minimum time to wait for the start of an ACK.
TsAck	Time to transmit an ACK.
TsCCAOffset	Start of slot to beginning of CCA.
TsCCA	Time to perform CCA.
TsRxTx	The longer of the time it takes to switch from receive to transmit or vice versa.

For validated messages, the destination device must inspect the destination address in the DLPDU. Under normal conditions, the destination device acknowledges all messages addressed to it. The acknowledgement consists of the device switching from receive mode to transmit mode and beginning its ACK such that the Start of Message (SOM) occurs exactly TsTxAckDelay after the end of the transmitted source device's packet.

Meanwhile the source device is turning around its transceiver by switching from transmit mode to receive mode. The source device must enter receive mode and be listening for communication by TsRxAckDelay after the end of its transmission. The source must listen for the ACK's SOM for a duration TsAckWait.

For an acknowledged transaction, the packet is successfully forwarded only when both the source packet and the ACK packet have been successfully received by the destination and the source device respectively.

Un-Acknowledged (Broadcast) Transmissions
Broadcast transmissions are also supported. In these messages the DLPDU source address is specific and the destination address is the broadcast address. Broadcast messages are not acknowledged at the Data-Link level.

The source device must begin its transmission such that the end of the Start of Message (SOM) occurs exactly TsTxOffset after the start of the slot (as described in Acknowledged Transactions) whether a CCA is performed or not.

The destination device must enter receive mode and be listening for communication by TsRxOffset from the start of the slot. The destination must listen for the SOM for a duration TsTxWait. If the destination device detects the SOM then it must receive and validate the message.

A message containing the broadcast destination address is never acknowledged and, consequently, this completes the transaction and the communications in the slot.

9.2 Communication Tables and Buffers

All devices maintain a series of tables that control the communications performed by the device and collect statistics on those communications. In addition, packets are buffered as messages are received, processed and forwarded.

The tables controlling communication activities include:

- Superframe and Link tables. Multiple superframes may be configured by the network manager. Multiple links within a superframe are configured to specify communication with a specific neighbor or broadcast communications to all listening to the link.
- The Neighbor table. The neighbor table is a list of all devices that the device may be able to communicate with.
- The Graph table. Graphs are used to route messages from their source to their destination. The device does not know the entire route rather, the graph indicates the next hop destinations legal for propagating the packet onward toward its destination.

In addition to these tables, there is a packet queue that buffers messages (see Subsection 9.2.5). Devices must support the minimum number of table entries shown in Table 4.

The communication tables and the relationships between them are shown in Figure 12. Within the device the neighbor table is central. This table contains a list of all devices that have been identified by the device.

Note: Although specific implementation of data and configuration storage is left up to the designers of the WirelessHART devices, general description of the fields contained in the data structures is important to overall understanding. Some fields described in the tables in this section may be calculated or derived from other information, and do not necessarily occupy space on the device.

A graph may specify more than one neighbor any of which may be used for the next hop for packets following the route designated by the Graph ID. In other words, when forwarding a packet using graph routing the device can propagate it to any of the neighbors associated with that packet's Graph ID. For more information see Subsection 9.3.

The device maintains network time synchronization and tracks the absolute slot number. These slots are organized into superframes. All communications are scheduled to occur within a superframe during specific slots in that superframe. All devices must support multiple superframes

Each superframe has one or more links. The links specify the slot and associated information required to forward or accept a packet.

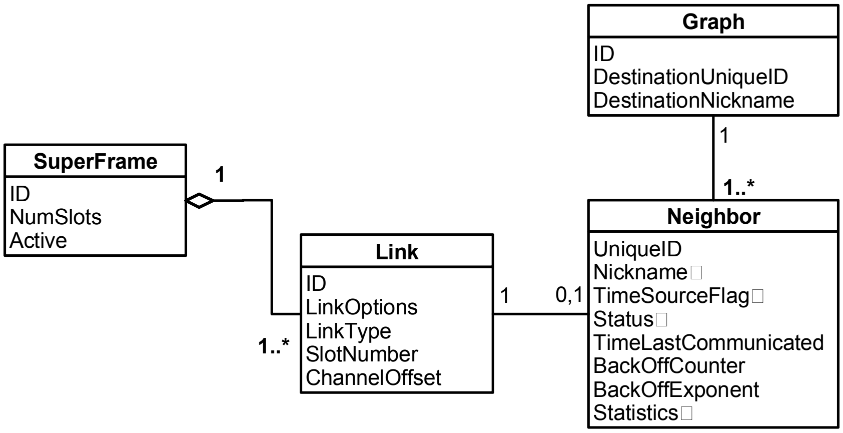


Figure 12. Data-Link Table Relationships

Each link defines a communications opportunity. A link can belong to one and only one superframe. Links specify a neighbor that is the partner in any communication using the link. A link is either directed to a neighbor or is a broadcast link. For a link, packets are either propagated to or from the neighbor. Broadcast links always propagate packets from the device.

In many cases the information in these tables is shared between the Network and Data-Link Layers (see the *Network Management Specification* for more information). The following subsections describe each of the Data-Link tables and buffers in more detail.

Table 4. Minimum Table and Buffer Space Requirement

Description	Minimum Number Required
Neighbors	32
Superframes	16
Total Number of Links	64
Graphs	32
Total Number of Graph-Neighbor pairs	128
Packet Buffers	16

9.2.1 Superframes

Each device must support multiple superframes. Superframes are created and maintained by the Network Manager. Superframes consist of a fixed number of slots (see Table 5), initially contain no links and begin as disabled.

Once a superframe is created the Network Manager adds, deletes and modifies links within the superframe, thus identifying opportunities for device to device communications. Once configured with links, the superframe can be enabled to allow the link scheduler to begin identifying transmit and receive slots.

Table 5. Superframe Properties

Content	Description
Unsigned-8 SuperframeId	Unique identifier of the superframe. This is supplied by the network manager.
Unsigned-16 NumSlots	Number of slots in the superframe (size of superframe)
Bits-1 ActiveFlag	Flag indicating if the superframe is currently activated
Links [ ]	List of links in this superframe

9.2.2 Links

Links are allocated to a superframe by the Network Manager. The link includes a reference to a neighbor that is permitted to communicate with the device. This reference can be to a single neighbor or the link can be a broadcast to (an unspecified) group of neighbors. Furthermore, the slot number within the superframe, direction of the communication (transmit/receive), link characteristics (e.g., shared/dedicated), and the initial communication channel are specified.

When the Network Manager designates a link as being shared, contention-based, multiple-sender access is performed within the corresponding link. For these links, messages that are not acknowledged result in a random back-off algorithm being applied (see Subsection 9.3.3).

Table 6. Link Properties

Content	Description
LinkId	Unique identifier of the Link.
Ref NeighborId	Reference to a Neighbor table entry.
Enum-3 LinkType	Indicates the type of link: { **normal, broadcast, join, discovery** }
Bits-1 TxLink	When set, indicates the link may be used for transmit
Bits-1 RxLink	When set, indicates the link may be used for receive
Bits-1 SharedLink	When set, indicates the link is shared by multiple devices
Unsigned-16 Slot	Slot number in superframe
Unsigned-6 ChannelOffset	Frequency hopping channel offset

Note: The Network Manager shall not delete or suspend any join links while there are outstanding received join requests.

Link Channel Calculation

The link also specifies the ChannelOffset and thus implicitly the channel hop order. For a given link and absolute slot, the actual channel used is determined by dividing the sum of the ChannelOffset and absolute slot number by the number of channels currently active. The remainder of this operation indexes the channel table to obtain the actual channel used for communication in the active slot. The channel table contents and length is Physical Layer dependent. For example, the IEEE STD 802.15.4-2006 (2450 MHz) Physical Layer supports 16 Channels (see Subsection 10.1).

To complete this channel calculation, each device must contain a 64-bit ChannelMap parameter that tracks the device's active channels. Each bit (bit0 through bit63) that is set identifies an active channel. The ChannelMap is initialized to all ones (i.e., all channels are active by default). Only bits corresponding to legal channels for the Physical Layer are significant and considered in any calculations.

Consequently, the active channel can be calculated using the modulo function:

ActiveChannel = (ChannelOffset + Absolute Slot Number) % Number of Active Channels

Once the ActiveChannel value is calculated the ChannelMap is used to find the channel used for the communication. The active and significant bits in the ChannelMap are organized into an array of bit numbers ("ActiveChannelArray").

ActiveChannelArray [ ] = { ordered set of active bit numbers in ChannelMap }

The bit number of the least significant active channel is placed at index zero in the array. The ActiveChannel value is used to index into the ActiveChannelArray.

Channel = ActiveChannelArray [ActiveChannel]

The result is the channel that must be used for communication in that Absolute Slot Number.

9.2.3 Neighbor Table

The device must maintain a list of neighbors it has knowledge of. These are companion devices that share a link with this device or neighbors whose communications have been overheard. As shown in Figure 12, the neighbor table is central to driving device communications:

- Each link has a reference to one neighbor (or it is broadcast link).
- Graphs may have references to several neighbors. When a graph is used for routing the list of neighbors held by the graph are all valid recipients of the packet being propagated.

The neighbor table entry collocates a variety of properties and statistics pertaining to the neighbor (see Table 7) including:

- Basic neighbor identity information;
- Performance and historical statistics; and
- Shared slot parameters;

Basic identity information includes the neighbor's Unique ID, 2-byte Nickname address and whether the device is a time source. To support contention-based access in shared slots, the neighbor table also contains the parameters necessary to support the back-off algorithm (see Subsection 9.3.3)

The device's ability to communicate with a neighbor is a key metric in forming and grooming the mesh network. Consequently, statistics are maintained in each neighbor table entry. These include average Received Signal Level (RSL); statistics on the packets transmitted and received and the timestamp of the last communication with the neighbor.

If a link to that neighbor exists, the LastTimeCommunicated is used to trigger transmission of Keep-Alive packets. A Keep-Alive must be transmitted to the neighbor (see Subsection 9.3) whenever the LastTimeCommunicated is greater than the keepAliveInterval. Keep-Alive transmissions are repeated until a new DLPDU is received from the neighbor.

The PathFailureTimer is also maintained in the Neighbor Table. Whenever a DLPDU from the neighbor is received, the timer is initialized to pathFailInterval. When this timer reaches zero, the PATH_FAILURE.indicate SP must be invoked. When this occurs, the timer is re-initialized to pathFailInterval and restarted. The device shall keep trying to use the failed path until the neighbor or its links are removed from the table.

Every packet that is received updates the corresponding neighbor table entry or creates a new one. Neighbors the device shares links with must be retained. Neighbors without links to this device can be deleted. When the neighbor table is full and a new neighbor is overheard, the neighbor with the oldest lastTimeCommunicated is deleted and the new neighbor is added to the table.

Table 7. Neighbor Table Entry

Content	Description
Unsigned-40 NeighborUniqueId	Unique ID of the neighbor device (i.e., the long address)
Unsigned-16 NeighborNickname	The short address of the neighbor
Unsigned-4 JoinPriority	Join Priority
Bits-1 TimeSourceFlag	Flag indicating if device should take time synchronization from this neighbor
Bits-7 Status	Status information regarding this neighbor (e.g., Path failure)
Unsigned-3 BOExp	Back-off exponent in collision avoidance algorithm for shared links
Unsigned-5 BOCntr	Back-off countdown in collision avoidance algorithm for shared links
Time LastTimeCommunicated	Time when last communicated with this neighbor
Time PathFailureTimer	Cyclical path failure timer. Resets to pathFailInterval after each successful communications. The PATH_FAILURE.indicate SP is invoked whenever PathFailureTimer reaches zero.
Signed-8 AvgRSL	Average received signal level (in dBm) for packets received from neighbor.
Unsigned-16 PacketsTransmitted	Number of (non-broadcast) packets transmitted to the neighbor
Unsigned-16 MissedAckPackets	Number of packets for which an expected ACK was not received
Unsigned-16 PacketsReceived	Number of good (non-broadcast) packets received from the neighbor
Unsigned-16 BroadcastsReceived	Number of good broadcast packets received from the neighbor

9.2.4 Graph

The graph provides the routing information to guide the delivery of a packet to its final destination. A graph is a directed list of paths that connect two devices within the network. Both upstream (toward the Gateway) and downstream graphs are used in WirelessHART. The Network Manager is responsible for correctly configuring each graph. Graphs have an ID; a list of neighbors and (optionally) the destination's long and short address.

At the original source device for a packet the upper layers identify the packet's final destination and the graph to use when routing the packet.

The list of neighbors identify those devices that are legal Data-Link destinations for the packet's next-hop toward its final destination. Since it is a possible for a graph to specify multiple next hops, redundancy and reliability is built into graph routing. Individual neighbor references are sometimes called a "connection".

Table 8. Graph Table Entry

Content	Description
Unsigned-16 graphId	Unique Graph Id.
Unsigned-40 destUniqueID	Destination node’s address.
Unsigned-16 destNickname	The short address of the neighbor
Ref Neighbor [ ]	List of references to neighbors that are the next hop toward the destination

The destination addresses are required by all devices sourcing data to a specific final destination. However, the addresses are optional since (technically) intermediate devices may be merely forwarding the packet along its route and not sourcing data to the same final destination.

9.2.5 Packet Buffer List

All devices must maintain a list of packet buffers. These buffers are used to receive, process and transmit packets. The record associated with a packet is indicated in Table 9. The Packet ID is used to reference the packet and is created when the packet is added via the TRANSMIT.request service primitive.

The PacketTimeStamp is set when the packet is added to the transmit list. The time stamp is used to select the packet to transmit when either of two equal priority packets can be propagated in the same absolute slot. In this case the older packet is sent first. Also, in worst case scenarios after a long time elapses, the time stamp can be used to automatically flush a very old packet.

In addition, the record contains the packet's priority and the specification of the packet's destination.

Table 9. Packet Record

Content	Description
PacketId	Unique Packet ID.
Payload	The Data-Link Payload (i.e. the NPDU)
Priority	Transmit priority of the packet
Destination	Graph, source or proxy routing information or broadcast destination.
PacketTimeStamp	Indicates when a packet was added to the transmit list

Packet Priorities

Since the device must be able to store multiple packets, pending their propagation, it is possible that multiple packets could be candidates for transmission in the same slot. When that happens, priority and the age of the packet are used to select the packet to be transmitted in that slot.

More specifically, when there are multiple packets that can be transmitted in the slot the highest priority packet is chosen for transmission (see Subsection 8.3). If multiple packets are tied for the highest priority then the oldest packet is transmitted first.

Destinations
The destination of a DLPDU can be specified as one of the following types:

- *Graph Route* - If the Network Layer specifies a Graph ID as the destination then the Data-Link can send the packet to any of the devices associated with that graph
- *Source Route* - If the Network Layer specifies a specific device address as the destination then the Data-Link must transmit the packet on a link to that neighbor. If the device does not have a link to that neighbor then the Network Manager must be notified that a Source Route Error has occurred.
- *Broadcast* - If the destination address is broadcast then the DLPDU must be transmitted using a broadcast link from the designated superframe.
- *Proxy Route* - When the Network Layer indicates the destination address is that of a joining device then the packet must be transmitted in a Join link.

The destination type determines the link type that can be used and ultimately the slot the in which the DLPDU can be transmitted.

9.3 Link Scheduling

All devices must maintain a link schedule that identifies the next slot that must be serviced. Servicing the slot consists of either listening for a new packet or propagating a packet onward through the mesh. When a slot has both a packet waiting to be propagated and receive links, propagating the packet shall have priority over attempting to listen for a new packet.

While, on the surface, link scheduling seems straightforward, it is complicated by (for example) transaction priorities, the modification of links, and the enabling and disabling of superframes. Each event that affects link scheduling may result in widespread reassignment of transmit links. For example, if a high priority transaction fails transmission then it must be rescheduled. Consequently, lower priority transactions may need to be deferred to a later link and their current link ceded to the higher priority transaction. Effects can be even more widespread, for example, should a superframe be disabled or even deleted.

Every event that can affect link scheduling must result in the link schedule being re-assessed (see Subsection 9.3.4). Link scheduling consists of evaluating the packets pending propagation and determining the first Absolute Slot Number that can be used to propagate a packet. Next, all receive links must be assessed to determine the first Absolute Slot Number that can be used in attempting to receive a new packet. The first slot, either transmit or receive, must be scheduled for servicing.

9.3.1 Servicing Transmit Links

Figure 13 summarizes the potential relationships that affect the calculation of the next transmit slot to be serviced. Packets received from the Network Layer are scheduled for a slot based on the graph, the destination address and whether or not the device is acting as a proxy for a device joining the network. Fundamentally, each packet has a destination and a priority. As specified in Subsection 9.2.5, there are several types of destinations, each of which affect slot selection. For each type of destination the set of transmit links must be determined:

- *Dest-Graph* - For a graph-routed destination, the set of links shall be all transmit Links for all Neighbors in the Graph.
- *Dest-Broadcast* - If the destination address is broadcast the set of links shall be all broadcast Links for the designated Superframe ID.
- *Dest-Neighbor* - When a single neighbor is specified as the destination, the set of links shall be all transmit Links to that single Neighbor.
- *Dest-Proxy* - If the destination address belongs to a joining device then the set of links shall be all transmit Links of type Join.

Once the set of links suitable for transmitting the packet are determined, the set of upcoming slots must be determined. The intersection of the upcoming slots and the links with a pending packet determines the next slot to be scheduled for propagating a packet.

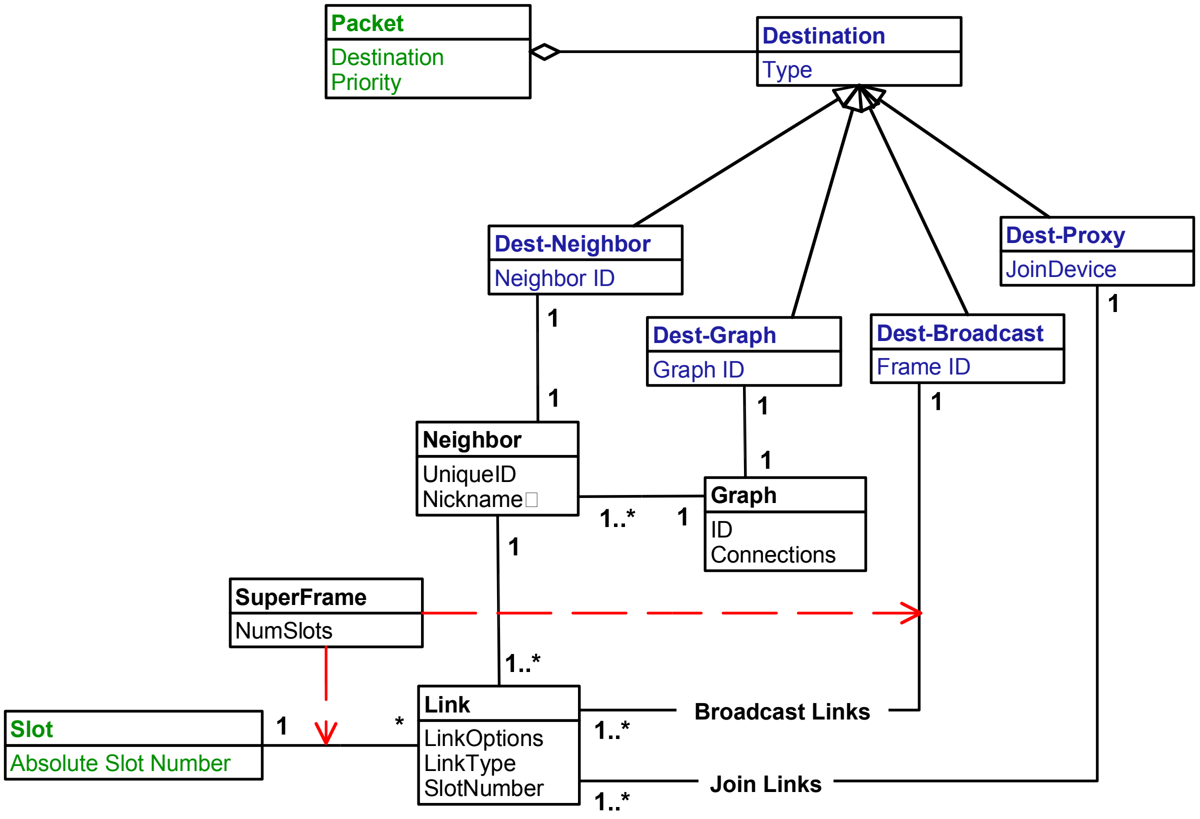


Figure 13. Relationships Used for Link Scheduling

The Absolute Slot Number is used to track the occurrence of each slot. This huge integer indicates the number of slots since the network was originally formed. Links that are soon to occur within a superframe can be identified:

SuperframeSlot = (Absolute Slot Number) % Superframe.NumSlots

Using this information an ordered list of the links that are about to occur on all slots can be constructed.

> Note: Links may be shared (see Subsection 9.3.3), used to advertise (see Subsection 9.3.4) the device's presence to new devices attempting to join the network or used to discover neighbors (see Subsection 9.3.5). The back-off algorithm in shared slots must be employed to determine when a transmission in a shared slot is allowed. Advertise packets (see Subsection 9.3.4) are transmitted periodically as specified by the Network Manager.

Links that cannot be used to propagate a pending packet are ignored. From the resulting set of links, the first potential link and its associated slot can be identified. When there are multiple packets that can be propagated in the slot, the rules in Table 10 shall be used to select the packet. The rules are applied top-to-bottom and the evaluation stops as soon as a single packet is identified.

Table 10. Packet Precedence Order

Tie-Breaker Number	**Rule** (Apply Top-to Bottom until a single packet is identified)
0	Choose the packet(s) with the highest priority
1	Choose the packet destined to the neighbor communicated with longest ago.
2	If keep alive time has expired with a neighbor, generate a keep alive packet to the neighbor with communicated with longest ago.
3	If an advertise time has lapsed (see Subsection 9.3.4) then generate an Advertise packet.

Once the rules have been applied, the next transmit slot and the packet to be transmitted has been identified.

9.3.2 Servicing Receive Links

For each active superframe, all receive links will be scheduled. The set of upcoming receive links can be calculated in the same fashion as with transmit links (see Subsection 9.3.1). The main difference is that all receive links should be serviced. Once the ordered list of links is created, the earliest slot is selected and the link within that slot with the lowest Superframe ID number becomes the candidate for servicing. If there are no other receive links to service then the device must service the Discovery receive link. In any case, this receive link will be serviced if there are no pending packets to be transmitted on or before this slot.

9.3.3 Shared Slots

Shared slots are assigned to many source devices one, or more of which may attempt to convey a packet within that slot and channel. Consequently, collisions may occur within a shared slot. If a collision occurs, the destination device will not be able to successfully receive any source's transmission and will not produce an acknowledgement to any of them. To reduce the probability of repeated collisions, source devices shall use random back-off delay when their transmission in a shared slot is not acknowledged (i.e., no ACK is received by the source device).

A device shall maintain two variables for each neighbor: Back-Off Exponent (BOExp) and Back-Off Counter (BOCntr). Both of these variables are initialized to 0. When a transaction in a shared slot fails the random back-off period is calculated based on the BOExp. For each unsuccessful attempt by the source device in a shared slot the BOExp is incremented and a sequential set of numbers calculated. The set of numbers consists of the whole numbers {0, 1, ... L} where

L = ((2 to the power BOExp) - 1)

The following table shows sample random back-off sets for BOExp values of one (1) to four (4).

Table 11. Example BOCntr Selection Sets

BOExp	Set of Possible Values for BOCntr
1	{ 0, 1 }
2	{ 0, 1, 2, 3 }
3	{ 0, 1, 2, 3, 4, 5, 6, 7 }
4	{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}

From this set calculated based on the BOExp, a random value for the BOCntr is selected. For each subsequent shared link to that neighbor, the BOCntr must be decremented. Only when the corresponding BOCntr is zero can the source device attempt a transmission in a shared slot.

Note: The value of BOExp shall not exceed that of MaxBackoffExponent

Since it is also possible that interference can cause packet loss, the back-off exponent and counter are maintained on a neighbor-by-neighbor basis. If communication with that neighbor fails on a dedicated link then the device must assume channel degradation (rather then a collision) caused the failure in the shared slot and the BOExp and the BOCntr shall both be reset to zero.

Broadcast messages must not be transmitted on shared slots.

9.3.4 Advertising

Nodes that are already part of the network may be configured by the Network Manager to advertise the network and facilitate joining of new devices. The AdvertiseInterval attribute sets the interval at which Advertise packets (see Subsection 8.2.4) are generated. Whenever the AdvertiseInterval lapses an Advertise packet shall be transmitted on the first available non-shared transmit link. When AdvertiseInterval is set to zero then an Advertise packet shall be generated whenever a non-shared transmit link is available.

An Advertise packet may be sent on any non-shared transmit link that is not in use. In general, all other traffic has higher priority than advertising. However if a the advertising timer has expired and the next transmit slot is already set aside for a Keep-Alive then an Advertise DLPDU to the designated neighbor must be sent instead of the Keep-Alive DPDU.

9.3.5 Neighbor Discovery

Devices must continuously listen for communications from their neighbor and for communications from new neighbors. Continuous monitoring of neighbors and the discovery of new neighbors is critical to the maintenance of the mesh and the enhancement of communications reliability.

Upon receiving any DLPDU, from a new neighbor the device must invoke the NEIGHBOR.indication SP along with that neighbors address and the DLPDU's RSL. Periodically the Network Layer communicates the list of new neighbors to the Network Manager. The device may receive a DLPDU, addressed to the device, from anew device attempting to join the network. When this happens the device must add the device to the neighbor table. Then the device must duplicate the join links and insert the joining device as the link neighbor. These neighbors are perishable and may be deleted based on the LastTimeCommunicated to make room for new neighbors (see Subsection 9.2.3). When a neighbor is deleted, the auto-created join links connecting the device to the joining device shall be deleted.

To aid in the discovery of new neighbors the device listens whenever possible on "Discovery" links. Discovery links are shared by all devices in the network. In addition to listening, a device must also randomly transmit a Keep-Alive message in a (transmit) Discovery Link. This allows other devices listening on the Discovery link the opportunity to discover the device.

The frequency at which the device transmits in the discovery link is bounded by the DiscoveryInterval. To schedule the discovery transmission, the device must select a random time period between 0 and DiscoveryInterval and use this to initialize the discovery timer. When that timer expires, the device must set a TimeToDiscover flag to TRUE and schedule the next randomly timed discovery transmission.

> Note: A DiscoveryInterval value of -1 (i.e., 0xFFFFFFFF) indicates discovery links are not serviced and discovery transmissions are not generated.

Once the TimeToDiscover flag is set, link scheduling will transmit the Keep-Alive packet at the first available discovery (transmit) link. The Keep-Alive packet should be addressed to the linked neighbor with the oldest LastTimeCommunicated value.

9.4 MAC Operation

This specification decomposes the MAC sub-layer into three primary components (see Figure 14):

- The TDMA machine that specifies the overall operation of the MAC sub-layer;
- The transmitter that sends a DLPDU; and
- The receiver that listen to a link and receives a DLPDU (should one be transmitted).

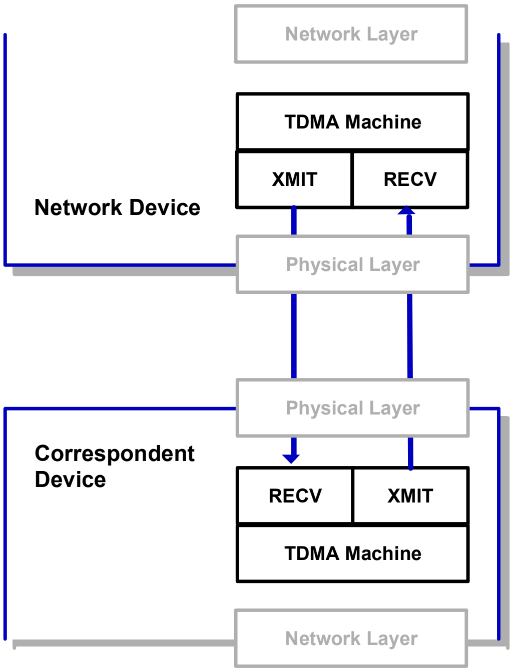


Figure 14. MAC Components

Device requirements are specified in this section and state transition diagram [Hatley] are used to clarify these requirements. The state transition diagram notation is summarized in the *Token-Passing Data-Link Layer Specification*.

9.4.1 TDMA Machine

The operation of the TDMA Machine is shown in Figure 15. Operation of the TDMA Machine begins when the device joins a network; is configured with a list of superframes, graphs and links; and begins receiving packets from other devices or from the device's Network Layer. Normal operation can be divided into three basic responsibilities:

- Managing schedules;
- Propagating DLPDUs to other devices and acquiring DLPDUs; and
- Maintaining time synchronization.

Managing schedules includes creation and maintenance of superframes, links, and neighbor statistics. Furthermore, as packets are acquired and propagated the schedules must be updated by invoking the "Schedule Link" process (see Subsection 9.2.5). Link scheduling consists of evaluating the active superframes, links and packets pending conveyance to identify the next slot that needs servicing.

The schedules determine the dispatching of DLPDUs. All receive links must be serviced by attempting DLPDU reception. Since links must be allocated to support possible retries, often there are more receive links than transmission links. Since transmission is often successful, many of the receive links will be unused, not containing a corresponding transmission. Most received DLPDUs contain packets destined for the Network Layer. Data-Link Layer command DLPDUs are destined for the Data-Link Layer.

After joining the network, the "Idle" state is entered. The following events can occur while in the Idle state:

- Slot Timeout. The most frequent activity performed by the TDMA Machine is the servicing of a SlotTimeout event. This event indicates a transmit or receive slot needs to be serviced. If the slot timed out to propagate a message (link type is transmit) then the "Talk" state is entered. Otherwise, the "Listen" state is entered.
- A modification to the device's list of superframes or links. Modifications to superframes (e.g., the enabling or disabling of one) or links (their addition or deletion) affects link scheduling. These changes typically result in link definitions being revised and may result in a different number of transmit and receive attempts per second.
- A FLUSH.request. This causes a packet to be discarded and may cause the slot timeout to be changed. Once the packet is discarded the FLUSH.confirm service is invoked.
- A TRANSMIT.request. This adds a packet to be propagated to the device's packet queue and may affect link scheduling.

All of these events require the link schedule to be re-evaluated and the next active slot to be identified.

Propagating a message

The device maintains a list of packets to be conveyed to one or more neighbors. When a transmit slot with a pending packet occurs (slot timeout and the link is transmit), the device will attempt to propagate the packet to its neighbor(s). These attempts will result in success or one of several negative outcomes.

- Successful propagation of a packet with a DLPDU destination address that is the broadcast address occurs as soon as the packet is transmitted. The packet's buffer can be released immediately upon completion of the DLPDU transmission.
- Successful propagation of a packet with a DLPDU destination address that is not broadcast address occurs when a validated, successful ACK is received. This indicates that message propagation was completed successfully, so the packet's buffer is released.
- If the ACK contains an error Response Code then the neighbor (specified by the link) has refused the packet (e.g., it does not have capacity to forward the received packet). When this occurs, the packet shall be retained and its propagation retried.

- If no response is received then the packet will be rescheduled and transmission retried. Should the PDU timeout the queued packet is returned to the Network Layer for disposition and possible re-routing.

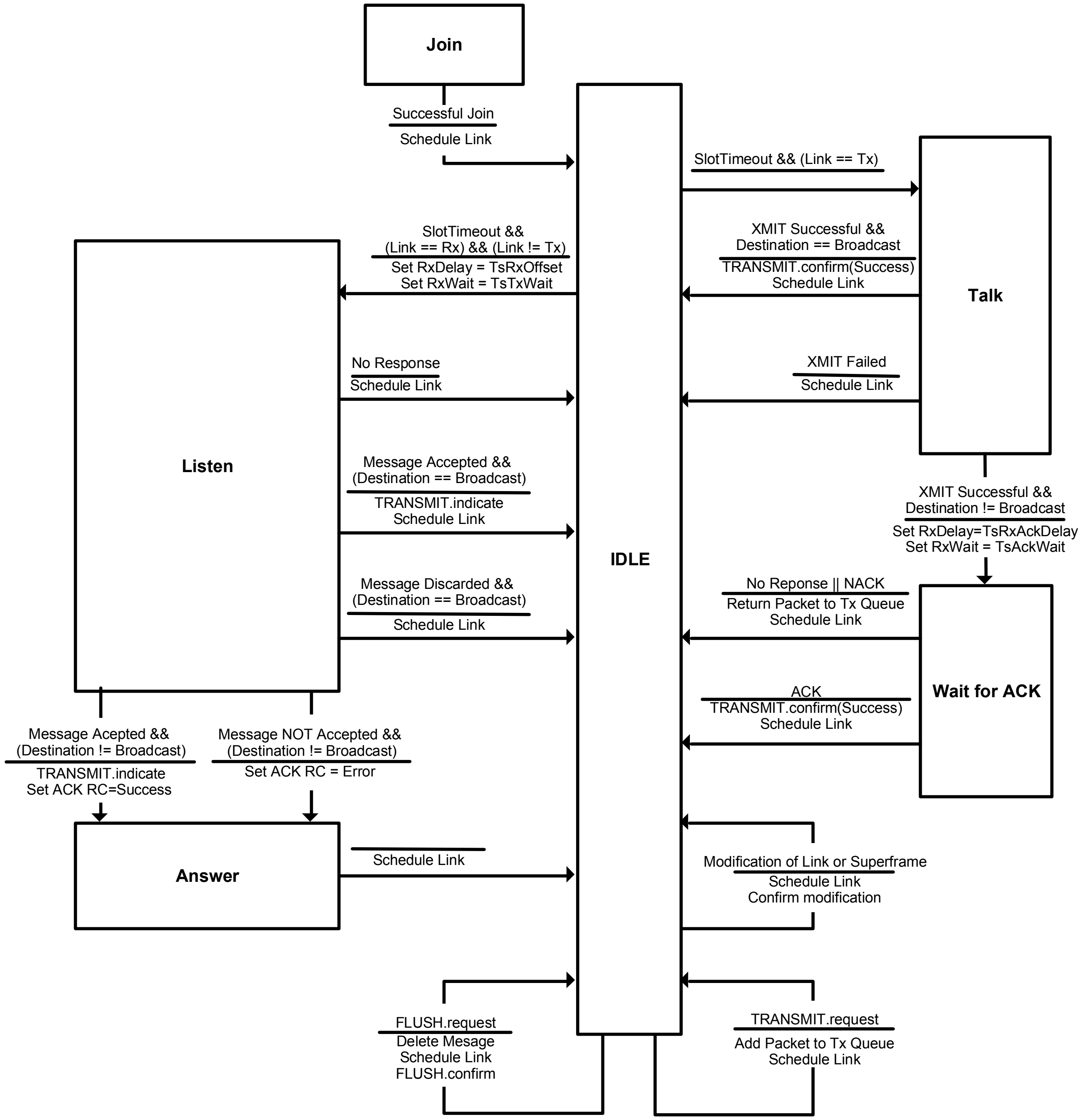


Figure 15. TDMA State Machine

Propagating a message consists of transmitting the packet and (optionally) receiving an ACK to confirm packet acquisition by the destination device. When a transmit slot with a pending packet occurs the "Talk" state is entered by invoking the XMIT engine (see Subsection 9.4.2) and the "Talk" state waits for its completion. The XMIT engine will successfully transmit the packet or, if CCA fails, the transmission attempt will fail. If the transmission fails, the transaction is aborted and link schedule is evaluated.

If the DLPDU destination is the broadcast address, there will be no ACK. The transaction is complete and the link schedule is evaluated.

Upon successful transmission and if the DLPDU destination is not broadcast, the TDMA Machine transitions to "Wait for ACK" by initializing the RxDelay timer to TsRxAckDelay and the receive window (RxWait) to TsAckWait and calling the RECV engine. The RxDelay timer allows the receiving device to process the received PhPDU, run its AES-128 cipher engine and authenticate the contained DLPDU. The RxWait timer is used to set the duration of the receive window.

The TDMA Machine stays in "Wait for ACK" until the RECV engine completes. If the RECV engine indicates there was "No Response" the transaction fails. If it was a shared link, the BOExp and BOCntr are reevaluated to produce a random back-off period before the next transmission attempt in a shared slot (see Subsection 9.3.3). If it was not a shared link, the BOExp and BOCntr are reset to zero. Then, after link schedule evaluation, the TDMA Machine transitions back to "Idle".

If an ACK containing the "Success" Response Code is received then communication was successful (i.e., the packet was successfully forwarded). However, if an error Response Code was received the neighbor did not accept responsibility for the packet (i.e., the packet was not forwarded). In either case, network time synchronization may be assessed using the Time Adjustment field in the neighbor's response. If the neighbor is a time-source for the device, then the device must resynchronize its network time using the Time Adjustment field. If the neighbor is not a time source then, no time correction is performed.

Receiving an ACK indicates the neighboring device has successfully received the message and accepted responsibility for it. This allows the packet buffer containing the transmitted DLPDU to be released. The TDMA Machine returns to "Idle" after evaluating the link schedule.

Acquiring a message

Unless there is a packet to transmit in the slot, all active receive links must be serviced as their slot occurs and the acquisition of a message must be attempted. The acquisition of a message has three possible outcomes: the message's final destination is the device itself and the message must be consumed; the message must be forwarded by the device toward the message's final destination; or the DLPDU is not addressed to the device. In all cases, when a message is acquired the corresponding neighbor table entry must be updated (or created if need be).

The DLPDU acquisition cycle consists of an attempt to receive a PhPDU, the validation of any PhLPDU received and, if the DLPDU destination address is not broadcast, the transmission of a response to a valid received DLPDU.

When attempting to acquire a packet the RxDelay timer is initialized to TsRxOffset and the receive window (RxWait) to TsTxWait. The TsTxWait time sets the duration of the receive window and represents the largest time drift between neighboring devices allowed by the protocol.

Once the receive parameters are initialized, the RECV engine is called and the "Listen" state is entered. From the "Listen" state the TDMA Machine transitions to the "Idle" or "Answer" state depending on the result returned by the RECV engine. If no packet was received the TDMA Machine evaluates the link schedule and returns to the "Idle" state.

If a packet was captured then network time synchronization is assessed. TsError is calculated by taking the difference between the actual start time and the predicted start time of the received PhPDU. If the neighbor propagating the packet is one of the device's time sources then the device must use the measured TsError to resynchronize its network time. If the neighbor is not a time source then, no time correction is performed.

While in the "Listen" state and upon successful DLPDU reception the device must decide whether to accept the packet or discard it. The device accepts or discards the packet based on the DLPDU priority, the current priority threshold (see Subsection 8.3) and the number of packet buffers currently occupied in the device. When a packet is accepted by the device it is either consumed by the Data-Link Layer itself or forwarded to the Network Layer for disposition.

Once the Data-Link Layer has determined DLPDU disposition (accepting or discarding the packet), the "Listen" state is exited as follows:

- If the DLPDU destination address is the broadcast address, then the TDMA Machine returns directly to the "Idle" state. If the packet is accepted then the TRANSMIT.indicate service is signaled. The link schedule is updated to calculate the next slot timeout.
- If the destination address is not the broadcast address and the packet was accepted, then an ACK (with Response Code = "Success") must be transmitted to the device propagating the message. The TRANSMIT.indicate service is signaled and the TDMA Machine transitions to the "Answer" state.
- If the destination address is not the broadcast address and the packet was not accepted then a ACK (with Response Code indicating the error) must be transmitted to the device propagating the message. After discarding the packet the TDMA Machine transitions to the "Answer" state.

DLPDUs received via communication over a link are either consumed by the MAC sub-layer (e.g., Keep-Alive DLPDUs) or their contained payload is delivered to the Network Layer.

If an ACK DLPDU is generated, the measured TsError is copied into the Time Adjustment field of the response DLPDU.

When an ACK is to be transmitted the "Answer" state is entered and an ACK transmission is performed as depicted in Figure 16.

Upon completing transmission of the ACK the TDMA Machine evaluates the link schedule and returns to the "Idle" state.

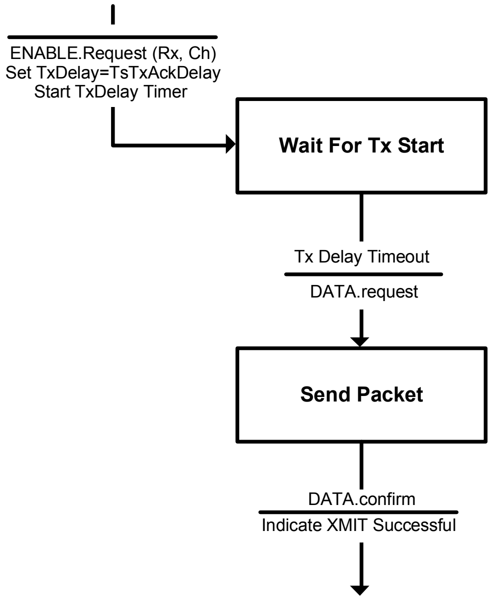


Figure 16. ACK Transmission

9.4.2 XMIT

When a DLPDU is to be transmitted, the XMIT engine is called to perform the actual DLPDU transmission (see Figure 17). This engine is called to propagate a message through either a dedicated or a shared slot.

Two basic transmit sequences are supported. The sequence to be performed is indicated by the CCAEnabled flag. The first sequence is a direct transmission (CCAEnabled is reset). The second sequence (CCAEnabled is set) consists of a Clear Channel Assessment (CCA) to verify the channel is not in use followed by the packet transmission.

- When CCAEnabled is set the Physical Layer is initialized using the ENABLE.request SP to place the transceiver into receive mode. TxDelay is set to TsCCAOffset.
- When CCAEnabled is reset the Physical Layer is initialized using the ENABLE.request SP to place the transceiver into transmit mode. TxDelay is set to TsTxOffset.

Next, the TxDelay timer is started. The TxDelay time is set to so that the transmission starts at the center of the neighbor's receive window. In other words, the goal of the transmitting device is for the PhL start delimiter to complete transmission at the center of the neighbor’s receive window. In addition, the device may use the TxDelay time to construct the DLPDU, run its AES-128 cipher engine and generate the MIC for the DLPDU. In some transceivers, the MIC can be generated as the PhPDU is transmitted.

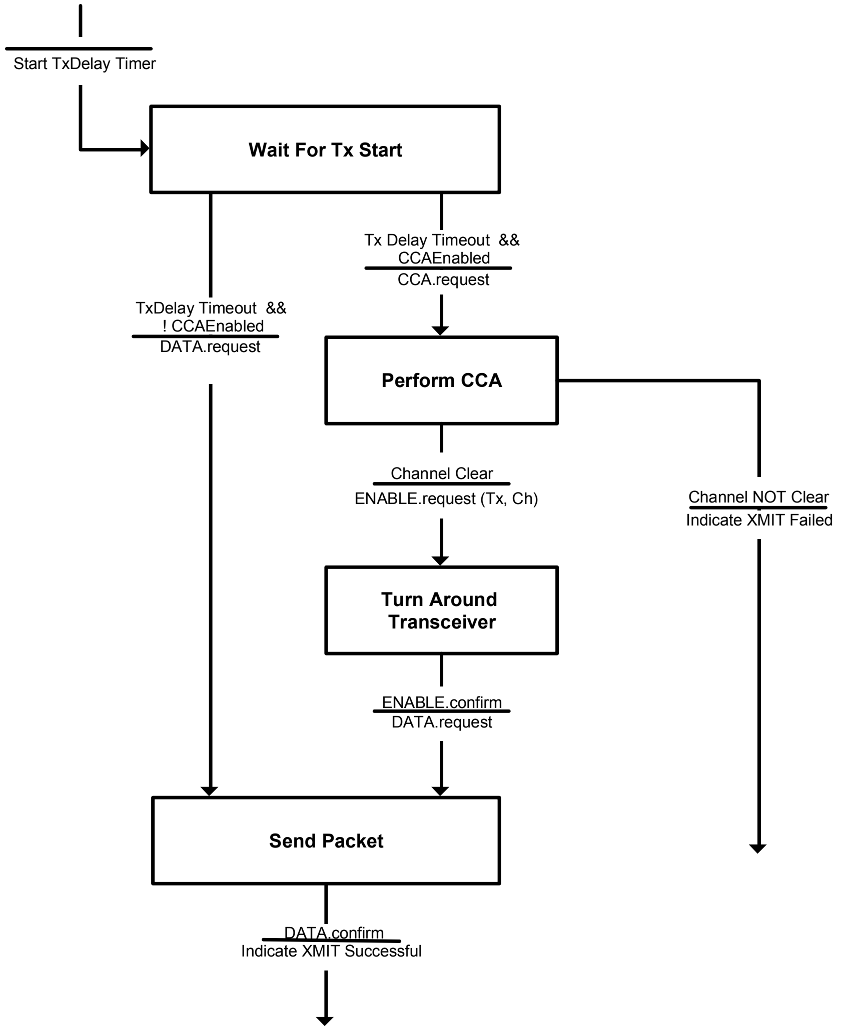


Figure 17. Transmit State Machine

Wait For Tx Start

Once initialization is complete, the "Wait For Tx Start" state is entered to defer start of transmission to the correct time within the slot. While in the "Wait For Tx Start" state, the channel is selected based on the link's starting channel offset and current absolute slot number. The transceiver mode (Rx or Tx) is selected based on the CCAEnabled flag.

Once TxDelay has timed-out, the XMIT engine transitions to the "Perform CCA" state if the CCAEnabled flag is set. Otherwise, the "Send Packet" state is entered.

Perform CCA

When the Perform CCA state is entered the Physical Layer primitive CCA.request is invoked and a CCA is performed. The state terminates when the CCA.confirm service is invoked by the Physical Layer. If the channel is clear the "Turn Around Transceiver" stated is entered.

If the channel is not clear then the XMIT engine terminates and indicated the transmission attempt failed (i.e., the packet was not transmitted).

Turn-Around Transceiver

Upon confirming the channel is clear the transceiver must be switched from receive to transmit mode using the ENABLE.request Physical Layer service. Upon reception of the ENABLE.confirm Physical Layer service the "Send Packet" state is entered.

Send Packet

Entering the "Send Packet" state, the packet transmission is started using the DATA.request Physical Layer service. Reception of the DATA.confirm event from the Physical Layer signals the completion of the transmission. The XMIT engine exits indicated a successful transmission.

9.4.3 RECV

All attempts to receive a packet are managed by the RECV engine (see Figure 18). This engine is called to acquire a message that is being propagated by one of the device's neighbors or during the device's message propagation sequence when awaiting an ACK.

Receive Attempt

When the RECV engine is called, the transceiver is configured by selecting the correct channel and placing the transceiver into receive mode using the ENABLE.request and ENABLE.confirm Physical Layer services. The channel is selected based on the link's starting channel offset and current absolute slot number. In addition, the RxDelay timer is started and the "Wait For Rx Start" state is entered. During the RxDelay the transceiver is allowed to settle and synchronize to the correct channel.

The "Wait for Rx Start" state is exited when the RxDelay timer lapses. The RxDelay timer is set by the TDMA Machine to allow the receiver to become active at the beginning of the receive window. The duration of the receive window is controlled by the RxWait timer which is started after the RxDelay timer lapses.

The transceiver is instructed to begin listening to the channel by invoking the DATA.request Physical Layer service primitive. The device stays in the "Listen for Packet" state until either 1) the start of a DLPDU (indicated by reception of the end of a PhPDU start delimiter) is detected or 2) the Rx Timer lapses and an RxTimeout occurs. If the RxWait timer lapses then the receive attempt fails and the RECV engine terminates indicating "No Response" and the communication statistics are updated accordingly.

If the expected start delimiter is detected, its time of arrival is recorded and the RECV engine transitions to the "Receive Packet" state to capture the balance of the packet. Upon receiving the DATA.indicate Physical Layer service primitive an initial evaluation of the received DLPDU is performed. If the DLPDU addresses are not as expected for the link the RECV engine terminates indicating "No Response" and the communication statistics are updated accordingly.

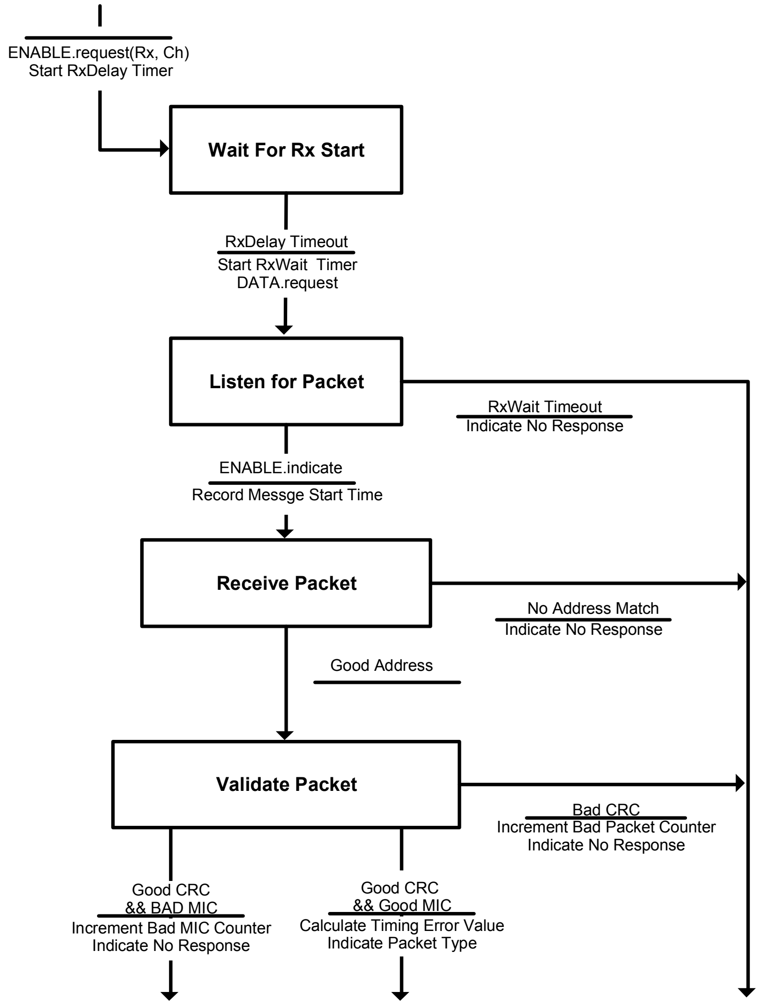


Figure 18. Receive State Machine

DLPDU Validation

If there are no addressing errors, the received DLPDU is validated. If the CRC is incorrect then the PhPDU was corrupted before or during reception, so the communication statistics are updated and the RECV engine terminates indicating "No Response".

If the received CRC is correct, the received keyed MIC is computed and checked. If the received MIC is not as expected, it may be indicative of an attack. Therefore, reception is considered a failure and both security and communication statistics are updated. The RECV engine terminates indicating "No Response".

If both the CRC and MIC verify, the reception is considered successful and the RECV engine exits indicating the type of DLPDU received.

10. PHYSICAL LAYER-SPECIFIC REQUIREMENTS

10.1 IEEE STD 802.15.4-2006 (2450 MHz) Physical Layer

This section specifies the requirements for devices supporting the IEEE STD 802.15.4-2006 Physical Layer. It also indicates the mapping of IEEE STD 802.15.4-2006 2450 MHz frequency channels to the indices used by the TDMA Data-Link Layer.

Table 12. 2450MHz IEEE STD 802.15.4-2006 Timing and Specifications

Symbol	Description	Value
	Data rate	250 kbit/s
	Symbol rate	62.5 ksym/s ± 40 ppm
TsTxOffset	The time between beginning of slot and start of packet transmission	2120 µs ±100 µs
TsRxOffset	Start of the slot to when transceiver must be listening.	1120 µs ±100 µs
TsRxWait	The time to wait for start of message.	2200 µs ±100 µs
TsMaxPacket	Maximum packet length (includes PhL header and DLPDU, i.e., 133 bytes)	4256 µs
TsTxAckDelay	End of message to start of ACK.	1000 µs ±100 µs
TsRxAckDelay	End of message to when transceiver must be listening for ACK.	900 µs ±100 µs
TsAckWait	The minimum time to wait for start of an ACK	400 µs ±100 µs
TsAck	ACK (26 bytes)	832 µs
TsCCAOffset	The time between start of slot and beginning of CCA operation	1800 µs ±100 µs
TsCCA	CCA detection time(8 symbols)	128 µs
TsRxTx	TxRx turnaround(12 symbols)	192 µs

Table 13. Physical Channel Table

Index	802.15.4 Channel	Frequency (MHz)
0	11	2405
1	12	2410
2	13	2415
3	14	2420
4	15	2425
5	16	2430
6	17	2435
7	18	2440

Index	802.15.4 Channel	Frequency (MHz)
8	19	2445
9	20	2450
10	21	2455
11	22	2460
12	23	2465
13	24	2470
14	25	2475
15		(Not Used)

The channel map array is two bytes long for IEEE STD 802.15.4-2006 (2450 MHz) Physical Layer. The least significant byte contains channels 0-7 from Table 13 and the most-significant byte contains channels 8-15 (bit 15 is always reset).

ANNEX A. REVISION HISTORY

A.1. Changes from Revision 1.0 to Revision 1.1

The changes in this revision include adding an addendum and reformatting the front page of the document to reflect the new HCF logo.

A.2. Revision 1.0

Initial Revision