IEC/PAS 62591

Edition 1.0 2009-01

PUBLICLY AVAILABLE SPECIFICATION

PRE-STANDARD

Industrial communication networks – Fieldbus specifications – WirelessHART™ communication network and communication profile

IEC/PAS 62591:2009(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub
The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC/PAS 62591

Edition 1.0 2009-01

PUBLICLY AVAILABLE SPECIFICATION

PRE-STANDARD

Industrial communication networks – Fieldbus specifications – WirelessHART™ communication network and communication profile

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE XH

ICS 25.040.40 ; 35.100.05

ISBN 978-2-88910-811-4

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

WirelessHART™ communication network and communication profile

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

A PAS is a technical specification not fulfilling the requirements for a standard, but made available to the public.

IEC/PAS 62591 has been processed by subcommittee 65C: Industrial networks of IEC technical committee 65: Industrial-process measurement, control and automation.

NOTE Use of some of the associated protocol Types in the IEC 61158 series are restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual property rights made by the holders of those rights permits a particular Data-Link layer protocol Type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol Types in other combinations may require permission from their respective intellectual property right holders.

IEC draws attention to the fact that it is claimed that compliance with this publication may involve the use of patents. IEC takes no position concerning the evidence, validity and scope of these patent rights.

The text of this PAS is based on the following document:

This PAS was approved for publication by the P-members of the committee concerned as indicated in the following document

Draft PAS	Report on voting
65C/506A/PAS	65C/513/RVD

Following publication of this PAS, which is a pre-standard publication, the technical committee or subcommittee concerned may transform it into an International Standard.

This PAS shall remain valid for an initial maximum period of 3 years starting from the publication date. The validity may be extended for a single 3-year period, following which it shall be revised to become another type of normative document, or shall be withdrawn.

INTRODUCTION

This IEC/PAS 62591 provides the specification, definitions, and profile of a future standard covering additions to IEC 61158 and additions to IEC 61784-1.

IEC 61158-5-20 Ed. 1 and IEC 61158-6-20 Ed.1 contain the application layer. This document adds the following:

– data link layer protocol specification;
– data link layer service definitions;
– fieldbus profile addendum for the data link layer specifying the wireless physical layer.

This document does not provide the required structure of the IEC 61158 series (for example separation of DL-service definitions and DL-protocol specification) and of the IEC 61784-1 series. The required structure will be provided during the process of becoming an International Standard.

The Type 20 protocol supports two-way digital communications for process measurement and control devices. Applications include remote process variable interrogation, cyclical access to process data, parameter setting and diagnostics. This document defines the specification that comprises the Type 20 field communications protocol for wireless devices. Specification of the Type 20 protocol is based largely on the OSI 7-Layer Communication Model (see Figure 1).

OSI Layer	Function	WirelessHART
Application	Provides the User with Network Capable Application	Command Oriented, Predefined Data Types and Application Procedures Auto-Segmented transfer of large data sets
Presentation	Converts Application Data Between Network and Local Machine Formats	
Session	Connection Management Services for Applications	Secured Session between Network Devices
Transport	Provides Network Independent, Transparent Message Transfer	Reliable stream transport, Negotiated Segment Sizes
Network	End to End Routing of Packets. Resolving Network Addresses	Power-Optimized Redundant Path, Mesh to the edge Network
Data Link	Establishes Data Packet Structure, Framing, Error Detection, Bus Arbitration	Secure & Reliable, Time Synced, TDMA/CSMA, Frequency Agile with ARQ
Physical	Mechanical / Electrical Connection. Transmits RawBit Stream	2.4GHz Wireless, 802.15.4 based radios, 10dBm Tx Power

Figure 1 – OSI 7-Layer Communication Model mapped to Type 20

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

WirelessHART™ communication network and communication profile

1 Scope

1.1 General

The Data-Link layer provides basic messaging communications between devices in an automation environment.

The protocol and services provide a communication for WirelessHART[1] based on a Physical layer according to IEEE 802.15.4.

1.2 Specifications

This PAS specifies

a) the procedures for the transfer of data and control information from one Data-Link user entity to another user entity or many, and among the Data-Link entities forming the distributed datalink service provider;

b) the structure of the fieldbus DLPDUs used for the transfer of data and control information by the protocol of this publication, and their representation as physical interface data units.

1.3 Procedures

The procedures are defined in terms of

a) the interactions between DL-entities (DLEs) through the exchange of fieldbus DLPDUs;

b) the interactions between a DL-service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives;

c) the interactions between a DLS-provider and a Ph-service provider in the same system through the exchange of Ph-service primitives.

1.4 Applicability

These procedures are applicable to instances of communication between systems which support communications services within the Data-Link layer of the OSI or fieldbus reference models, and which require the ability to interconnect in an open systems interconnection environment.

Profiles provide a simple multi-attribute means of summarizing an implementation's capabilities, and thus its applicability to various communications needs.

1.5 Conformance

This PAS also specifies conformance requirements for systems implementing these procedures. This standard does not contain tests to demonstrate compliance with such requirements.

---

[1] WirelessHART™ is the trademark of HART® Communication Foundation (HCF). HCF is a non-profit trade organization to support the HART Communication. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the registered trademark. Use of the trademark requires permission of the trade name holder.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61158-5-20:2007, *Industrial communication networks – Fieldbus specifications – Part 5-20: Application layer service definition – Type 20 elements*

IEC 61158-6-20:2007, *Industrial communication networks – Fieldbus specifications – Part 6-20: Application layer protocol specification – Type 20 elements*

IEC 61784-1:2007, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61804-3, *Function blocks (FB) for process control – Part 3: Electronic Device Description Language (EDDL)*

IEEE 802.15.4-2006, *Telecommunications and information exchange between systems-Local and metropolitan area networks-- Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)*

HART *Communication Foundation Common Tables Specification* HCF_SPEC-183, available at <http://www.hartcomm2.org/hcf/services_tools/doc_sales.html>

3 Terms, definitions, abbreviated terms, acronyms, and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
absolute slot number
count of all slots that have occurred since the network was formed and always contains the number of the current slot

NOTE The Absolute Slot Number is only incremented and is never to be reset.

3.1.2
acknowledge
explicit Data-Link response to the successful reception of a directed, non-broadcast DLPDU from a Data-Link source device and the second DLPDU of a two-DLPDU transaction

3.1.3
antenna gain
apparent power gain resulting from the antenna capability of concentrating power in a given direction

3.1.4
assailant
device generating interference

3.1.5
broadcast
sending of packets to all Network Devices that overhear the transmission

3.1.6
byte
8-bits; sometimes called an Octet

3.1.7
channel
RF band used to transmit a modulated signal carrying packets

3.1.8
channel blacklisting
method of eliminating a RF channel from usage

3.1.9
channel hopping
regular change of transmit / receive frequency to combat interference and fading

3.1.10
channel offset
link-specific value provided by the Network Manager that is used to calculate the channel to use when channel hopping

3.1.11
clear channel assessment
used to avoid initiating a transaction while the RF channel is in use. It is performed by listening to the channel prior to sending the first DLPDU of a transaction. If signal is detected the transaction is deferred to a later TDMA slot

3.1.12
coexistence
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 802.15.4-2006]

3.1.13
connection
data structure associated with graph routing that contains an ordered pair of Network Devices

3.1.14
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

3.1.15
Device Id
device nickname that uniquely identifies a device within a WirelessHART Gateway. A Client uses the Device Id to interact with the interfaces provided by the WirelessHART Gateway

3.1.16
device specific document
document provided by the supplier of a device providing details for a given device

3.1.17
discovery
method to locate or identify WirelessHART Gateways without human interaction

3.1.18
frame
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

3.1.19
frequency channels
allocation of the frequency spectrum in a given frequency range

3.1.20
gateway
network device containing at least one host interface (such as serial or Ethernet), acting as ingress or an egress point

3.1.21
graph
routing structure that forms a directed end-to-end connection between network devices

3.1.22
Graph Id
identifier used to indicate a specific graph entry

3.1.23
handheld
host application residing on a portable device

3.1.24
HCF Enumeration
number of data structures use enumerations which are controlled by the HART Communications Foundation (HCF). HCF maintains current lists of these enumerations

3.1.25
hop
movement of a packet directly between two adjacent neighbors in one network transaction without the participation of any other nodes in the network. Multiple hops are used to lengthen the transmit distance, bypass interference sources or avoid obstructions

3.1.26
interoperability
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

3.1.27
join
process by which a Network Device is authenticated and allowed to participate in the network. A device is considered joined when it has the Network Key, a Network Manager Session and a normal (not join) superframe and links

3.1.28
latency
time it takes for a packet to cross a network connection, from sender to receiver

3.1.29
lease
lease is an agreement between the host and the WirelessHART Gateway to share a resource for a future period of time; after which the resources can be reallocated for other purposes

3.1.30
link
full communication specification between adjacent devices in the network, that means, the communication parameters necessary to move a packet one hop. A link is a function of source/destination address pairing, slot and channel offset assignment, direction, (Tx/Rx or Rx/Tx), dedicated or shared communication, and link type. Links are assigned to Superframes as part of the scheduling process

3.1.31
link margin
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

3.1.32
Logical Link Control
higher of the two data link layer sublayers defined in the OSI Model. This sublayer handles error control, flow control, framing, and addressing

3.1.33
Medium Access Control
sub-layer found with the OSI Data-Link Layer (OSI Layer 2) used for arbitrating access to the communication channel

3.1.34
neighbor
adjacent nodes in the network such that the Receive Signal Level (RSL) suggests communication in at least one direction is possible

3.1.35
network device
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 Devices include Field Devices, Access Points (i.e. Gateways), Adapters, and Handhelds

3.1.36
Network Manager
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

NOTE 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.

3.1.37
node
addressable logical or physical device attached to the network

NOTE See also network device 3.1.35.

3.1.38
Nonce
number constructed so as to be unique to the current packet to ensure that old communications cannot be reused in replay attacks. The nonce is also necessary for maintaining packet secrecy and providing sender authenticity and packet integrity

3.1.39
omni-directional antenna
antenna with a radiation pattern that, when viewed from above, is equally strong in all directions

NOTE The antenna thus sends or receives signals equally well in all directions.

3.1.40
packet
generic reference to the set of data communicated across a network

3.1.41
packet error rate
average number packets (in percent) transmitted but not received correctly. For this specification, the reference PPDU for PER calculations is 20 bytes long (IEEE)

3.1.42
Peer
correspondent node at the other end of the communication link. The communication link terminates at the same protocol layer in the correspondent node

3.1.43
Physical Layer
Layer 1 in the OSI model is responsible for transmission of the raw bit stream and defines the mechanical and electrical connections and signaling parameters for devices

3.1.44
receiver sensitivity
minimum input signal required to produce a PER of less than 1 % with a PPDU 20 bytes long

[IEEE 802.15.4-2006]

3.1.45
Route Id
identifier used to indicate a specific route

3.1.46
service session
agreement between a Client and a WirelessHART Gateway that services shall be provided to the Client by the Gateway

3.1.47
Session Id
An identifier used to indicate a specific session entry

3.1.48
Shed Time
time between the last good message reception and the assumption of digital communication failure

3.1.49
slot
fixed time interval that may be used for communication between neighbors

3.1.50
superframe
collection of slots repeating at a constant rate. Each slot may have a link associated with it

3.1.51
Superframe Id
An Identifier used to indicate a specific superframe entry

3.1.52
throughput
effective data transfer rate of the network

3.1.53
time sequence diagram
graphical representation 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

3.1.54
time to live
field in the network header of each packet that specifies how many more hops a packet can travel before being discarded

3.1.55
transaction
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 a second, link-level acknowledgement DLPDU from the destination

3.1.56
unicast
sending of a packet to a single node in the network

3.1.57
UTF-8
8-bit UCS/Unicode Transformation Format
variable-length character encoding for Unicode. All XML described in this PAS is encoded using UTF-8 characters

3.2 Abbreviated terms and acronyms

ACK	See Acknowledge [3.1.2][2]
ADC	Analog-to-Digital Converter
ASN	See Absolute Slot Number [3.1.1]

[2] Figures in square brackets refer to subclause 3.1.

CCA	Clear Channel Assessment
DAC	Digital-to-Analog Converter
DAQ	Data Aquistion. This referes to a devices specific ADC or DAC
dB	Relative power decibels (3dB/octave, 10 dB/decade)
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	Data-Link Layer
DLPDU	Data-Link Protocol Data Unit (i.e., a Data-Link Layer packet)
DR	Delayed Response
DRM	Delayed Response Mechanism
DSSS	Direct Sequence Spread Spectrum
EEPROM	Electrically Erasable Programmable Read Only Memory. Non-volatile memory that is alterable by the Field Device without the use of external programming apparatus
EIRP	Equivalent Isotropic Radiated Power (also Effective Isotropic Radiated Power)
EUI-64	Extended Unique Identifier (64 bits long)
FSK	Frequency Shift Keyed
FTA	Field Termination Assembly (as referenced in Figure 6)
HCF	HART® Communication Foundation
ISM	Industry, Scientific, Medical Frequency bands
LoS	Line of Sight, an unobstructed distance between a transmitter and a receiver
LRV	Lower Range Value. Defines the relationship between a Dynamic Variable value and an analog channel lower endpoint (e.g. 4.00mA)
LSB	Least Significant Byte. The LSB is always the last byte transmitted over a HART data link
LTL	Lower Transducer Limit. The digital value that defines the minimum reliable and accurate value of a dynamic or Device Variable

MAC	See Medium Access Control [3.1.33]
MIC	Message Integrity Code
MSB	Most Significant Byte. The MSB is always the first byte transmitted over a HART data link
NAN	Not-a-Number
NPDU	Network PDU
O-QPSK	Offset - Quadrature Phase Shift Keying
OUI	Organizationally Unique Identifier
OUI	Organizationally unique identifier as defined by IEEE 802
PDU	Protocol Data Unit. The packet of information being communicated
PER	See Packet Error Rate
PhPDU	Physical Layer Protocol Data Unit (i.e. a Physical Layer packet)
PHY	See Physical Layer
PPDU	Physical Protocol Data Unit
RC	Response Codes
RSL	Received Signal Level. The signal level (in dBm) at a receiver input terminal
SP	Service Primitive
STX	Start of a transaction. An STX is used to convey a Network layer packet (an NPDU) from one node to an adjacent node
TDMA	Time Division Multiple Access
TER	Transaction Error Rate
TPDU	Transport PDU
TTL	Time To Live
UART	Universal Asynchronous Receiver Transmitter
URV	Upper Range Value. Defines the relationship between a Dynamic Variable value and an analog channel upper endpoint (e.g. 20,0 mA).
UTL	Upper Transducer Limit. The digital value that defines the maximum reliable and accurate value of a dynamic or Device Variable

WHA WirelessHART Adapter

WHD WirelessHART Device

4 Physical Layer

The Physical Layer is derived from IEEE 802.15.4. The profile is given in Clause 11.

5 TDMA Data Link layer

5.1 Purpose

Clause 5 defines the HART TDMA Data-Link Layer. This specification is applicable 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 devices. In other words, this publication specifies the rules used by HART devices to wirelessly communicate HART digital information. Figure 2 shows the scope of this specification.

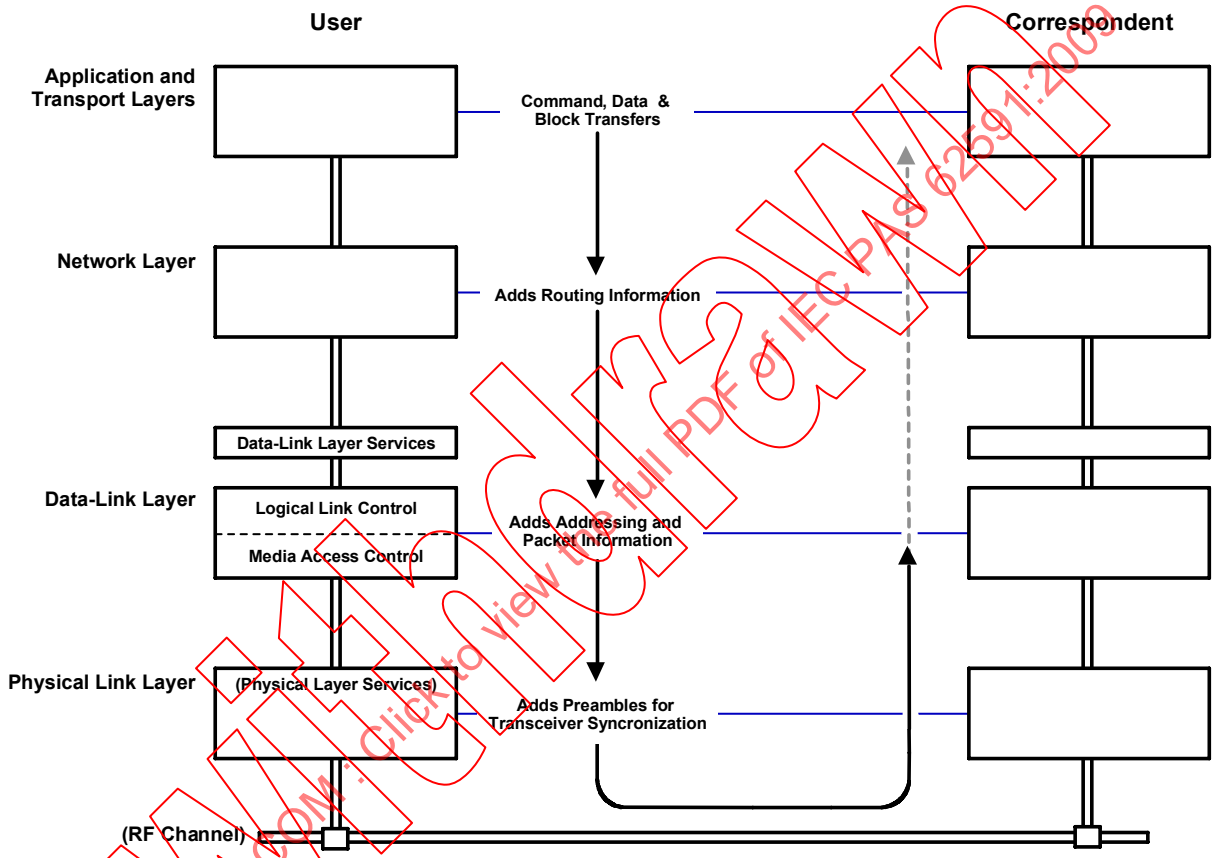


Figure 2 – Data-Link Layer Scope

Clause 5 includes the following:

- 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.
- 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.

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.

5.2 Overview

5.2.1 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 shall support multiple superframes, starting with superframe zero (0). At least one Superframe always enabled while additional superframes can be enabled or disabled (see Clause 6 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.

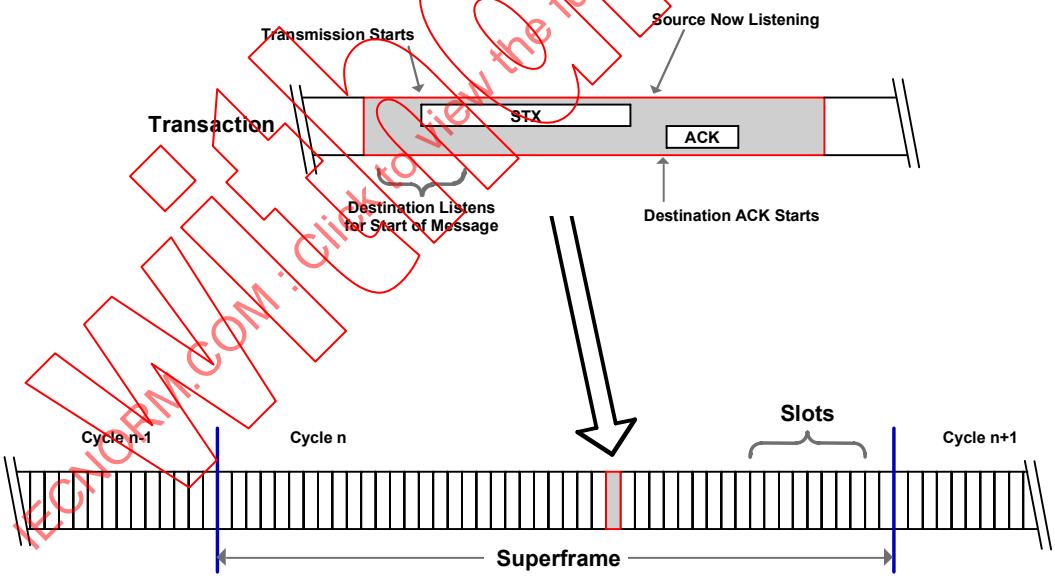


Figure 3 – A TDMA Slot and Superframe

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 shall contain either "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 Tables 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 shall 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 rendez-vous in time and frequency, thus promoting successful communications (see Figure 4).

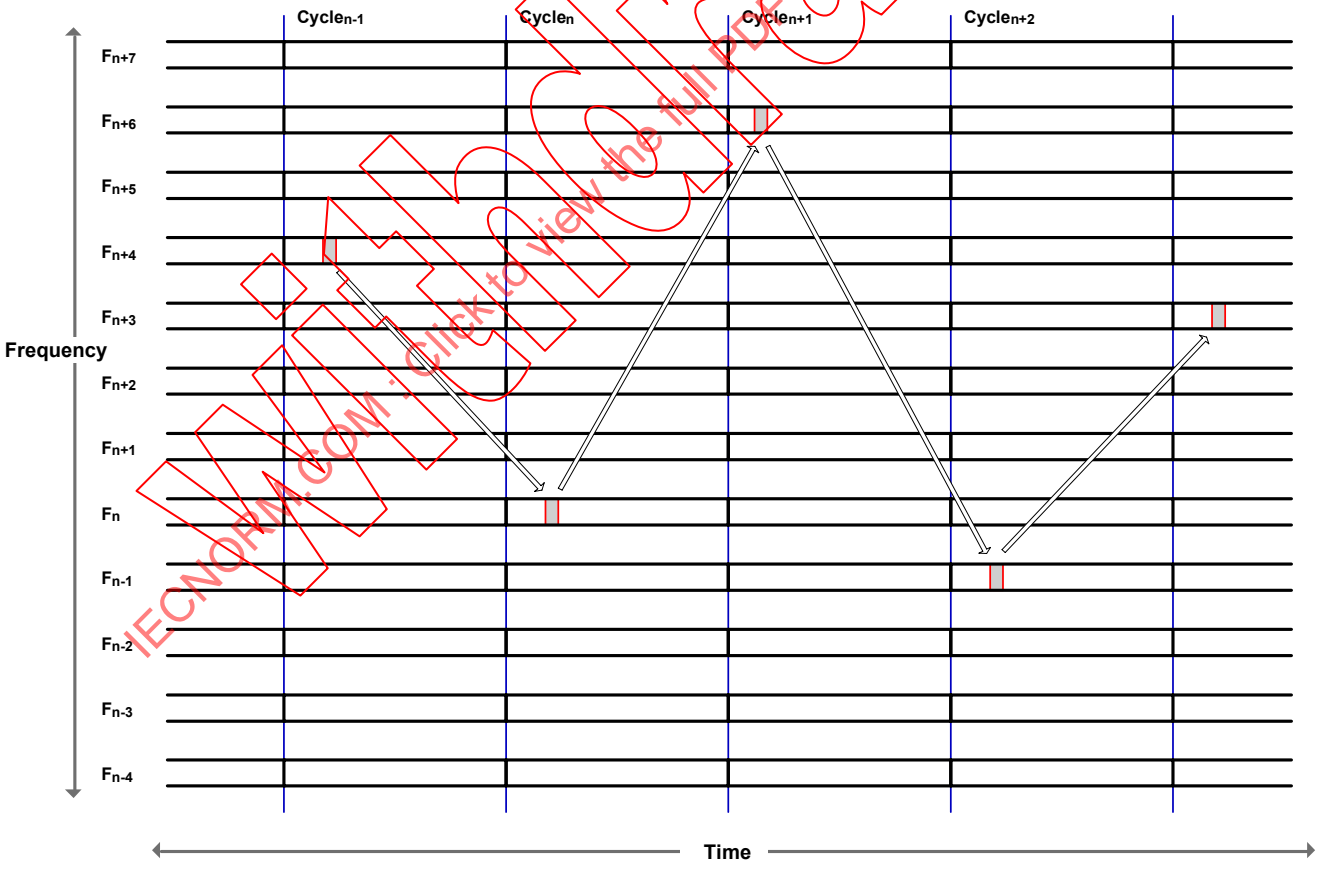


Figure 4 – Channel Hopping

Communicating devices are assigned to a superframe, slot, and channel offset. This forms a communication link between communicating devices. All devices shall 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 9 000 slots per superframe results in 135 000 possible links.

Channel hopping provides channel diversity, so each slot shall 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 determined by the Network Manager.

Assignment of links and the devices in a link is the responsibility of the Network Manager (see Clause 6).

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.

5.2.2 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, enabling a variety of applications (both high and low performance) to operate in the same network.

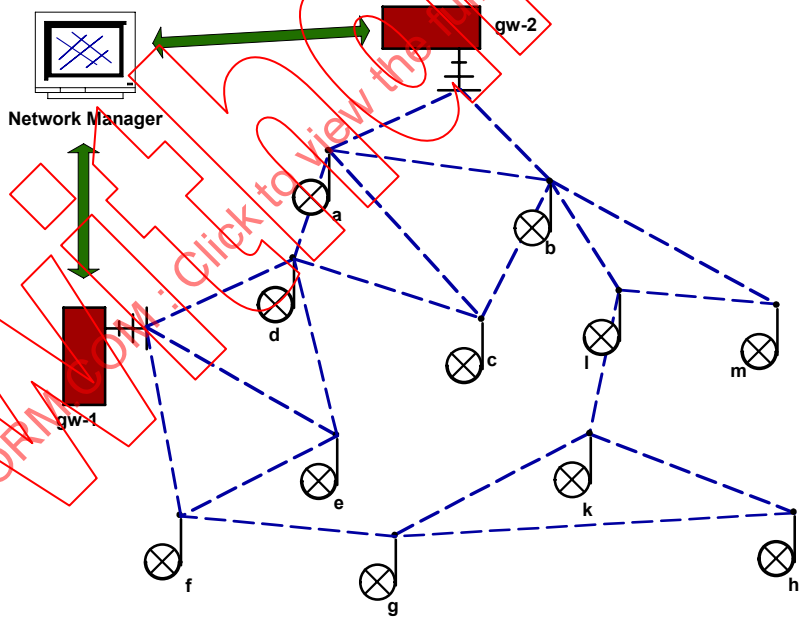


Figure 5 – Mesh Network

All network devices shall 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 can take several hops. The actual routing of packets is the responsibility of the Network Layer (see Clause 6).

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 shall be serviced by the device.

NOTE 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 (that means an ACK is received) to another device, the DLPDU is discarded.

5.2.3 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 shall keep its neighbor list current. 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 can become a neighbor of the advertising device. Further information about network joining can be found in Clause 6.

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 can be long intervals during which the link is not exercised. Keep-Alive DLPDUs are used to probe quiescent links and to maintain time synchronization.

5.2.4 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, voltage variations or ageing) is inevitable. Consequently, WirelessHART has several mechanisms to synchronize network-wide time.

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) shall 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 shall understand the time drift characteristics of their products. When the device's time source drifts, the device shall transmit Keep-Alive DLPDUs, as needed, to its time-synch neighbors to maintain time synchronization. Devices shall not require a Keep-Alive more often then once per 30 s while temperature is varying 2º C per min or less. Furthermore, device designs shall tolerate one retry in case of packet loss (i.e., a 10 s

safety margin). This corresponds to approximately a compensated clock accuracy of 10 ppm or better.

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

5.3 Data-Link Layer Services

5.3.1 General

Subclause 5.3 specifies the operation of the TDMA Data-Link layer from a "black box" point of view. Subclause 5.3 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 methodology.

The Services described in 5.3 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 shall 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 SPs. Optional parameters are distinguished by enclosing them within square brackets ("[","]") in the SP definitions.

5.3.2 Message SPs

5.3.2.1 General

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 initiates 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 scheduled 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 shall also allow multiple messages to be queued. In Figure 6, Sequence 2 illustrates this requirement.

NOTE 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 are queued for (possibly) 3 different Data-Link destinations.

Both unicast/directed (acknowledged), and multicast/broadcast (un-acknowledged transactions are supported. In Figure 6, Sequence 3 illustrates a multicast transaction.

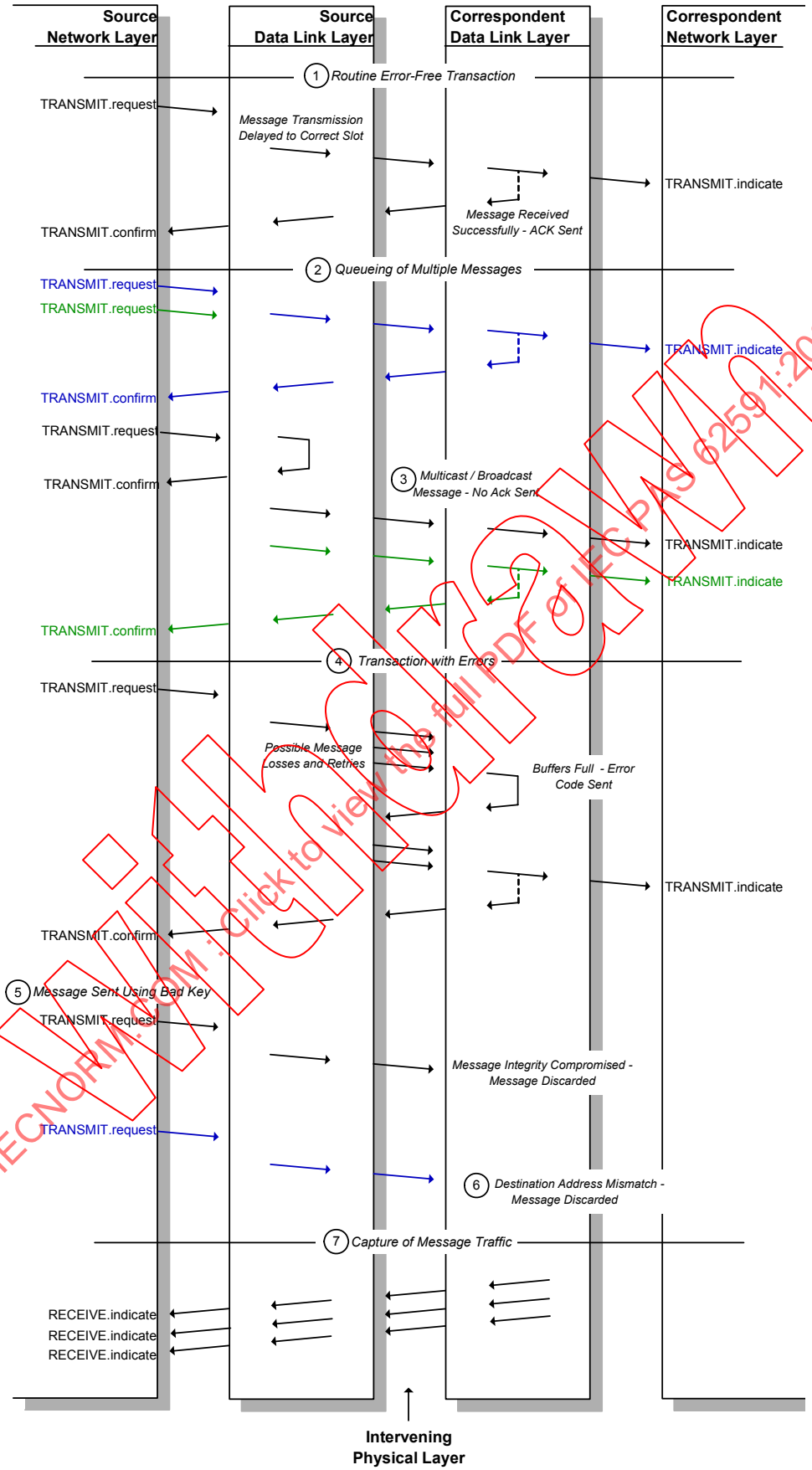


Figure 6 – Message Service Sequences

The Protocol also supports automatic retries to ensure reliable data exchange (see 5.5). A transaction including retries is shown in Figure 6, Sequence 4. 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 can result in no received response from the correspondent, or in receipt of an ACK with a Response Code indicating an error. If the 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 can also receive messages with either an invalid message integrity code (see Figure 6, Sequence 5) or a destination address that does not match the correspondent's (see Figure6, Sequence 6). These messages can be discarded and, in all cases, shall not be answered.

Figure 6, Sequence 7 shows the optional receive-only SP. This is available in a Data-Link Layer that supports a promiscuous operating mode where communications from other devices are captured.

5.3.2.2 Transmit SPs

5.3.2.2.1 TRANSMIT.request

```
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 shall be capable of queuing multiple requests and transmitting each packet on the appropriate link. The parameter to this SP include the following:

— ***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 element 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 5.4.3 for more information.

— ***timeout*** - The maximum time to attempt packet transmission. The Network Layer shall set this based on the ASN Snippet (see Clause 5).

— ***graph*** - This parameter is only present if graph routing is to be employed. When employed, the graph indicates the neighbors that can 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 shall be broadcast on the indicated Superframe.

— ***shortDestAddress*** - This parameter indicates the Nickname of the destination device that shall be used for the next hop.

— ***longDestAddress*** - This parameter indicates the Unique ID of the destination device that shall be used for the next hop.

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

5.3.2.2.2 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.

5.3.2.2.3 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.

5.3.2.2.4 FLUSH.request (handle)

Deletes the indicated packet.

5.3.2.2.5 FLUSH.confirm (handle, localStatus)

Indicates whether the packet was deleted.

5.3.2.3 Network Event SPs

5.3.2.3.1 DISCONNECT.indicate (localStatus, sourceAddress)

Notifies 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.

5.3.2.3.2 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.

5.3.2.3.3 ADVERTISE.indicate (localStatus, AdvertisePayload)

The SP is generated upon reception of an Advertise DLPDU (see 5.4.2.5). Upon receiving an Advertise packet, the device that is 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.

5.3.2.3.4 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.

5.3.2.4 Receive SPs

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

5.3.2.4.1 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.

5.3.3 Management SPs

5.3.3.1 General

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 Clause 5 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 can be accessed long after the Field Device has been on-line. For example, the Application Layer can receive a command from a network manager that changes the slots to be used when communicating.

5.3.3.2 LOCAL_MANAGEMENT.request (service, [data])

This SP is used to configure Data-Link Layer properties. The parameters Service and Data are defined in Table 1 hereinafter.

5.3.3.3 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.

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

This SP is used to notify `LOCAL_MANAGEMENT` of an un-requested MAC-sublayer event report, see Table 1.

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	

Service	Data	Description
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
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 shall 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	

Service	Data	Description
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 shall advertise Lower number indicates a better choice for joining
	Unsigned-4 joinPriority	
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 } MaxBackoffExponent defaults to 4
	Unsigned-4 MaxBackoffExponent	
WRITE_MAX_BACK _OFF_EXPONENT		
	Unsigned-4 MaxBackoffExponent	
ADD_CONNECTION		Adds a new connection to a device with specified graphs
	Unsigned-16 connectionhandle	Handle for this connection
	Unsigned-16 grasphId	graphID for the connection
	Unsigned-16 node Nickname	Address of device being connected via the specified graph
DELETE_CONNECTION		Deletes an existing connections
	Unsigned-16 connectionhandlet	Handle of an existing connection

5.4 Logical Link Control

5.4.1 The DLPDU

5.4.1.1 General

Subclause 5.4 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
- The 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.

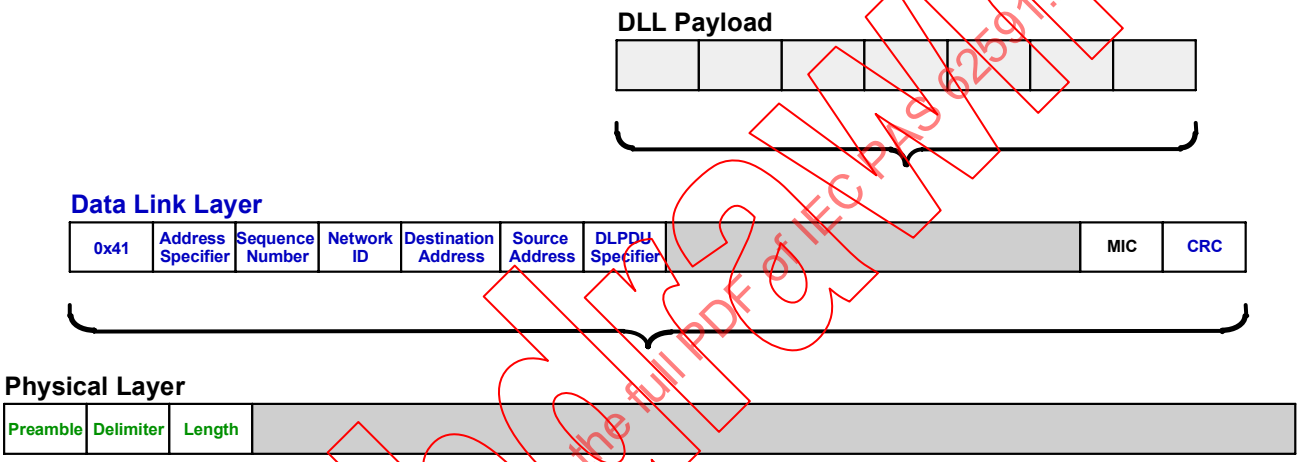


Figure 7 – DLPDU Structure

5.4.1.2 The Sequence Number

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

5.4.1.3 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
00 000-32 767 (0x0000-0x7FFF)	Permanent User defined networks (Critical networks)
32 768-36 863 (0x8000-0x8FFF)	Temporary User defined networks (Demos, trade shows, field trials, etc.)
36 864-57 343 (0x9000-0xDFFF)	Reserved
57 344-61 439 (0xE000-0xEFFF)	Manufacturing networks (non-public used by device manufacturers)
61 440-65 535 (0xF000-0xFFFF)	Reserved

5.4.1.4 The 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 can be used in a DLPDU. The other bits shall be set as indicated in Figure 8.

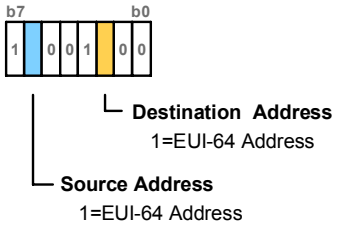


Figure 8 – Address Specifier

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 specify the broadcast address (i.e., 0xFFFF).

The EUI-64 address consists of a 3 byte "Organizationally Unique Identifier" (OUI) and the 5-byte Unique ID (controlled by the HART Protocol).

NOTE 1 The value of this central administrative number OUI is given by the IEEE Registration Authority Committee. Available at <http://standards.ieee.org/regauth>

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.

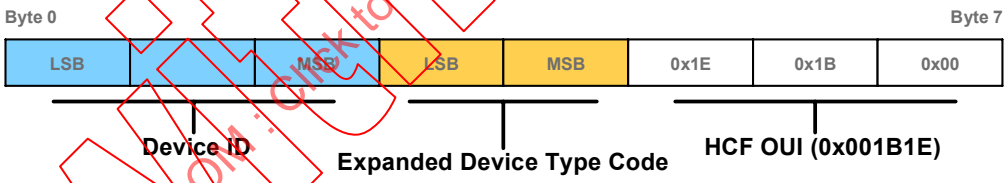


Figure 9 – Construction of 8 byte EUI-64 Addresses

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. Each device manufactured with the same Device Type Code shall 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 2 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).

5.4.1.5 The 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 shall mask off the most significant two bits. Devices built before any such future use is assigned shall set these bits to zero on transmission.

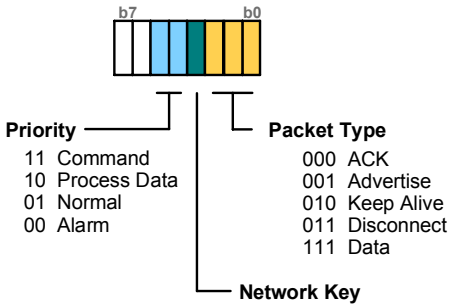


Figure 10 – DLPDU Specifier

The next two bits (i.e., bits 4 and 5) indicate the priority of the message (see 5.4.3). 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 shall set this bit and use the confidential Network-Key when generating the DLPDU MIC (see 5.4.4 for more information). This bit shall only be reset during the joining process (see Clause 5) 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.

5.4.1.6 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.

5.4.1.7 Keyed Message Integrity Code (MIC)

A keyed Message Integrity Code (MIC) is used for link-layer authentication of DLPDUs (see 5.4.4 for more information). Devices shall reply only to unicast, non-acknowledgement DLPDUs that have been successfully authenticated.

5.4.1.8 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.

NOTE At receivers, the received byte stream is subjected to a similar calculation. When a receiver may 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.

5.4.2 DLPDU Types

5.4.2.1 General

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 as follows:

- 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 shall 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 DLL command packets originated within the source Data-Link and consumed by a neighboring, destination peer Data-Link.

5.4.2.2 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.

5.4.2.3 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 shall 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 5.4.4 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 shall 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 shall 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 shall 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 µs. 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 µs. 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

5.4.2.4 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 can be used for the following purposes:

- 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.

5.4.2.5 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 supported 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 2 450 MHz 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 from the perspective of the joining device. 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 HCF Enumberations)
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

Byte	Format	Description
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. Shall 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

An Advertise packet can 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 5.4.4 for more information).

5.4.2.6 Disconnect DLPDUs

Disconnect DLPDUs are generated by devices leaving the network. This means the device is no longer available for communication and shall 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 5.4.4 for more information).

5.4.3 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 shall 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 shall 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 shall always be received, accepted and generate an ACK with a "Success" response code.
- If any buffers are available, DLPDUs received with "Command" level priority shall always be accepted and either consumed or forwarded. At least one buffer shall 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 shall be discarded. Furthermore, if the device does not have packet buffers available for that DLPDU, it shall 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.

5.4.4 Error Detection Coding and Security

5.4.4.1 General

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 5.4.1.8).

5.4.4.2 MIC Calculation

5.4.4.2.1 General

A keyed MIC is used to ensure that the DLPDU is 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', 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'.

5.4.4.2.2 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.

NOTE The Network Manager may change the network key from time to time.

5.4.4.2.3 DLL Nonce

The 'N' byte-string shall 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 and shall 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.

5.4.4.3 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.

5.5 Medium Access Control

5.5.1 General

The primary objectives of the Medium Access Control (MAC) sublayer are to maintain slot synchronization, identify slots that shall 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 the following:

- Tables of neighbors, superframes, links, and graphs that configure the communication between the device and its neighbors (see 5.5.3). These tables are normally populated by the Network Manager. In addition the neighbor table is populated as neighbors are discovered.
- A link scheduler (see 5.5.4) 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 5.5.5) 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.

5.5.2 Slot Timing

5.5.2.1 General

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's 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 shall enter receive mode. The device shall 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.

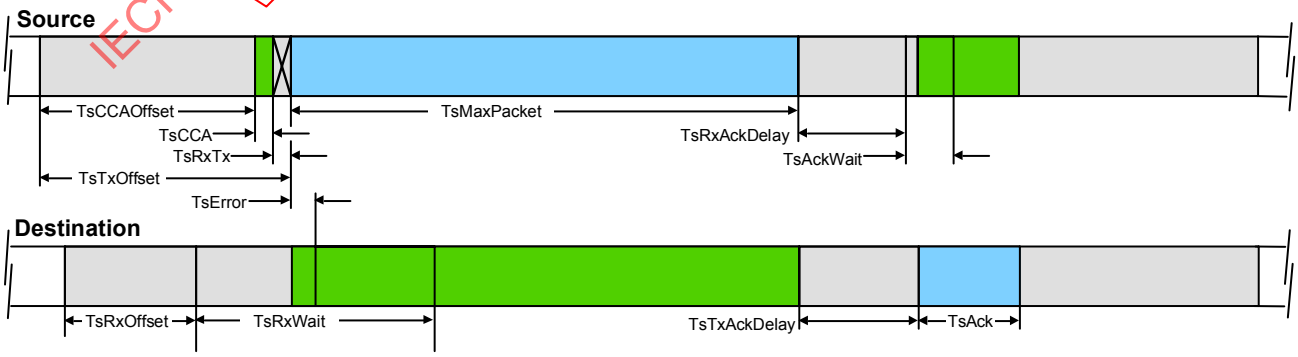


Figure 11 – Slot Timing

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.

5.5.2.2 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 shall contain a specific device address (i.e. not a broadcast address).

The source device shall 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 shall enter receive mode and be listening for communication by TsRxOffset from its start of the slot. The destination shall listen for the SOM for a duration of TsRxWait. If the destination device detects the SOM then it shall receive and validate the message. Any message that cannot be validated shall 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 shall 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 shall validate the STX, and generate an ACK during this interval, see note
TsRxAckDelay	End of message to when transceiver shall 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
NOTE Broadcast messages are not acknowledged.	

For validated messages, the destination device shall 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 shall enter receive mode and be listening for communication by TsRxAckDelay after the end of its transmission. The source shall 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.

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

5.5.3 Communication Tables and Buffers

5.5.3.1 General

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 the following:

- 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 5.5.3.6). Devices shall 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 5.5.3 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 5.5.4.

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 shall 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.

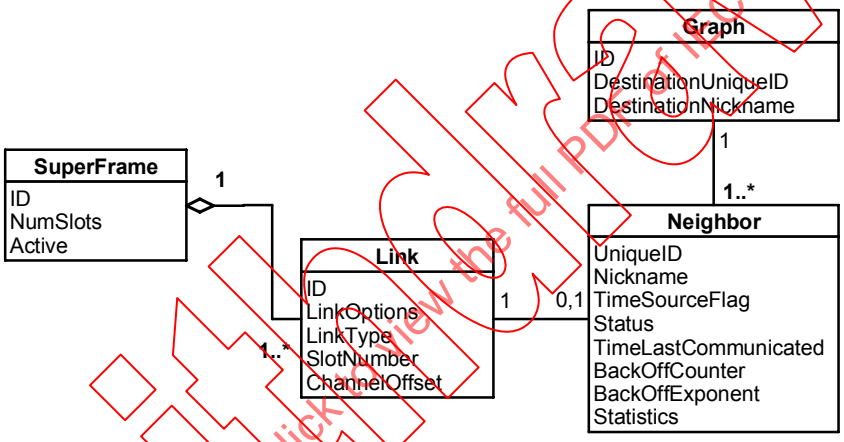


Figure 12 – Data-Link Table Relationships

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 Clause 5 for more information). Subclause 5.5 describes 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 Connections	128
Packet Buffers	16

5.5.3.2 Superframes

Each device shall 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

5.5.3.3 Links

5.5.3.3.1 General

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 5.5.4.4).

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.

5.5.3.3.2 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 (2 450 MHz) Physical Layer supports 16 Channels (see 13.3.1.3.1).

To complete this channel calculation, each device shall 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 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 shall 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.

5.5.3.4 Neighbor Table

The device shall 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 its 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 5.5.4.4).

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:

$$RSL = RSL - (RSL / RSLDamp) + (MeasuredRSL / RSLDamp)$$

Where MeasuredRSL is the RSL for the current packet and RSLDamp is the damping factor. RSLDamp shall be a power of 2 and defaults to 64. For discovered 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 shall be transmitted to the neighbor (see 5.5.4) 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 shall 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, shall 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-8 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

5.5.3.5 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. When the Graph ID value is less than 0x0100 it indicates a Frame ID, and when equal to 0xFFFF, it indicates the Graph is Invalid.

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 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.

5.5.3.6 Packet Buffer List

5.5.3.6.1 General

All devices shall 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 hereinafter. 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

5.5.3.6.2 Packet Priorities

Since the device shall 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 5.4.3). If multiple packets are tied for the highest priority, then the oldest packet is transmitted first.

5.5.3.6.3 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 shall transmit the packet on a link to that neighbor. If the device does not have a link to that neighbor then the Network Manager shall be notified that a Source Route Error has occurred.
- *Broadcast* - If the destination address is broadcast, then the DLPDU shall 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 shall be transmitted in a Join link.

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

5.5.4 Link Scheduling

5.5.4.1 General

All devices shall maintain a link schedule that identifies the next slot that shall 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 e.g, 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 shall 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, if a superframe is disabled or even deleted.

Every event that can affect link scheduling shall result in the link schedule being re-assessed (see 5.5.4.5). 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 shall 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, shall be scheduled for servicing.

5.5.4.2 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 5.5.3.6, there are several types of destinations, each of which affect slot selection. For each type of destination the set of transmit links shall 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 shall 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.

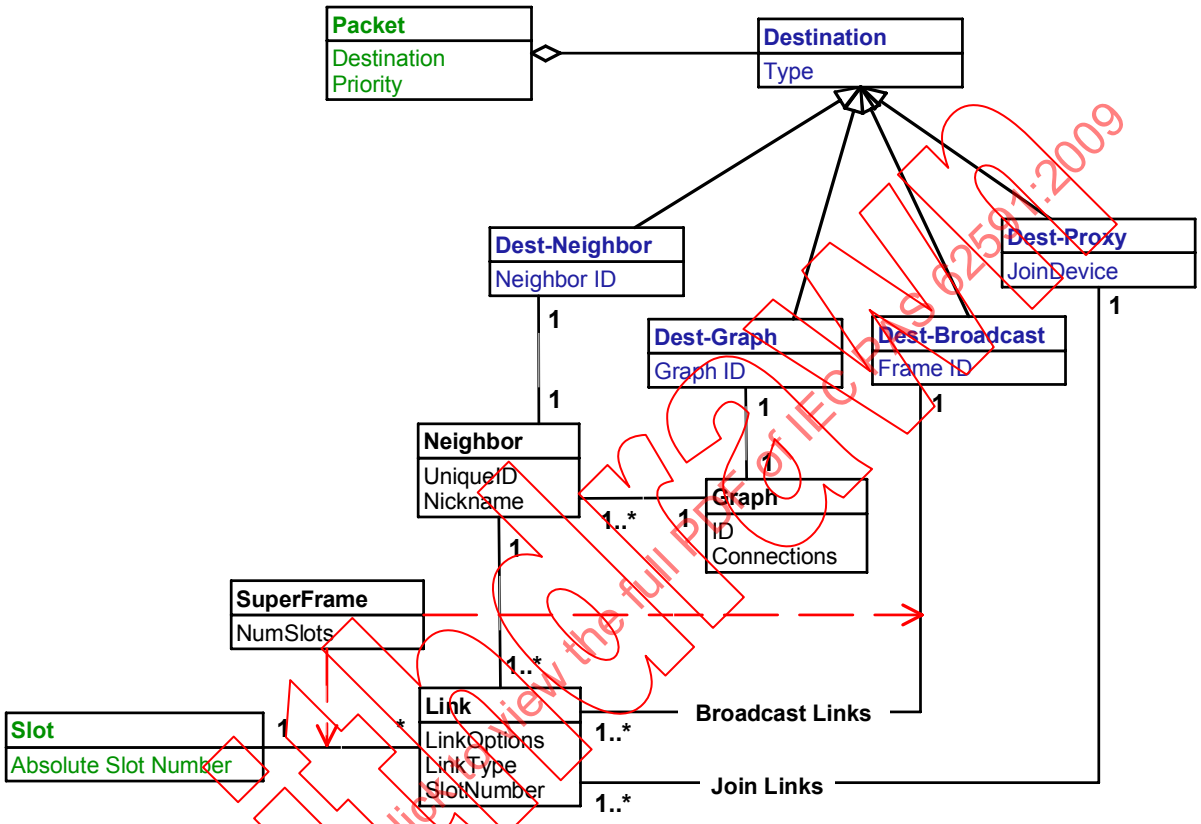


Figure 13 – Relationships Used for Link Scheduling

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 as follows:

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 5.5.4.4), used to advertise (see 5.5.4.5) the device's presence to new devices attempting to join the network or used to discover neighbors (see 5.5.4.6).

The back-off algorithm in shared slots shall be employed to determine when a transmission in a shared slot is allowed. Advertise packets (see 5.5.4.5) 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 5.5.4.5), then generate an Advertise packet

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

5.5.4.3 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 5.5.4.2). The main difference is that, baring the need to propagate a packet, all receive links shall 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 shall 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.

5.5.4.4 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)

Table 11 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}
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}

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 shall 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 shall assume channel degradation (rather than a collision) caused the failure in the shared slot and the BOExp and the BOCntr shall both be reset to zero.

Broadcast messages shall not be transmitted on shared slots.

5.5.4.5 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 5.4.2.5) 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.

5.5.4.6 Neighbor Discovery

Devices shall 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 shall 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 a new device attempting to join the network. When this happens the device shall add the device to the neighbor table. Then the device shall 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 5.5.3.4). 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 shall 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 shall select a random time period between 0 and DiscoveryInterval and use this to initialize the discovery timer. When that timer expires, the device shall 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.

5.5.5 MAC Operation

5.5.5.1 General

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

- 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 (if one is transmitted).

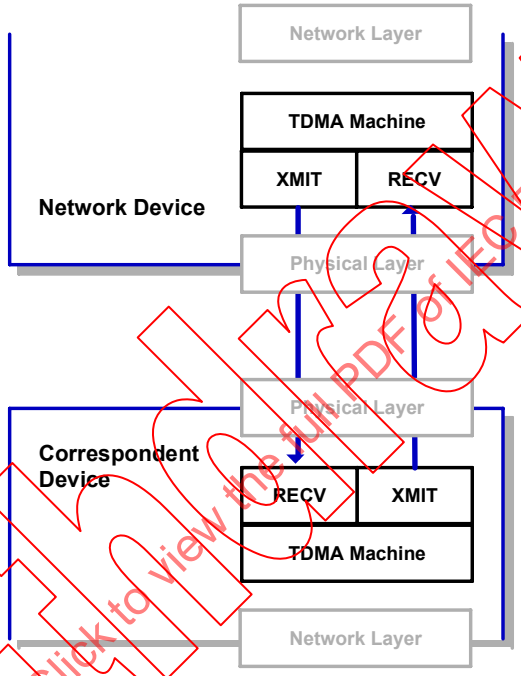


Figure 14 – MAC Components

Device requirements are specified in 5.5.5 and a state transition diagram [Hatley] is used to clarify these requirements. The state transition diagram notation is summarized in the Token-Passing Data-Link Layer Specification.

5.5.5.2 TDMA Machine

5.5.5.2.1 General

The operation of the TDMA Machine is shown in Figure 15 hereinafter. 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 shall be updated by invoking the "Schedule Link" process (see 5.5.3.6). 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 shall be serviced by attempting DLPDU reception. Since links shall 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.

5.5.5.2.2 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 the 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. If the PDU times-out the queued packet is returned to the Network Layer for disposition and possible re-routing.

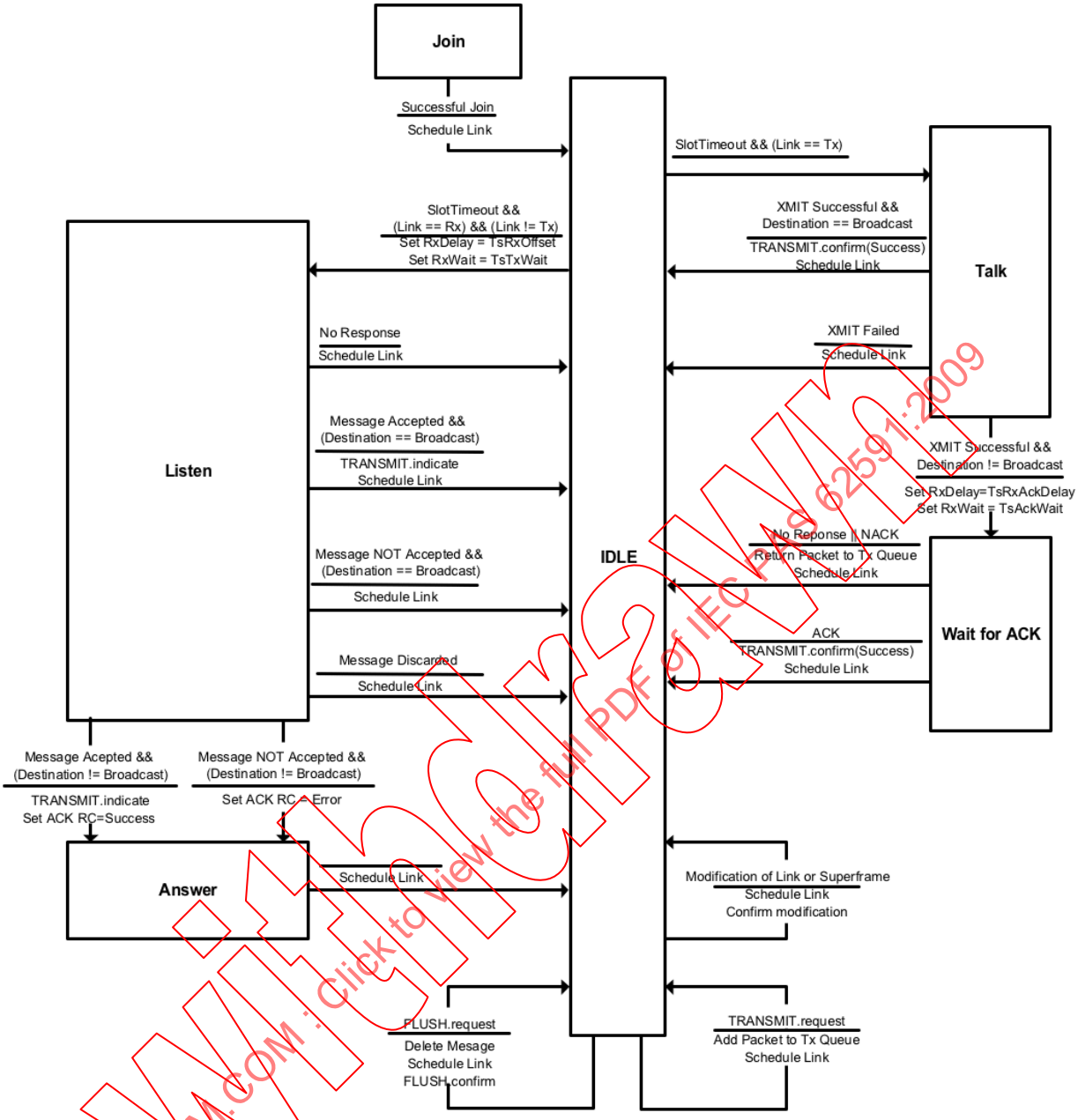


Figure 15 – TDMA State Machine

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 5.5.5.3) 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 5.5.4.4). If it was not a shared link, the BOExp and BOCntr are reset to zero. Then, after link schedule evaluation, the TDMA Machine transitions sets 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 shall 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.

5.5.5.2.3 Acquiring a message

Unless there is a packet to transmit in the slot, all active receive links shall be serviced as their slot occurs and the acquisition of a message shall be attempted. The acquisition of a message has three possible outcomes:

- the message's final destination is the device itself and the message shall be consumed;
- the message shall 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 shall 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 shall 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 shall 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 5.4.3) 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") shall 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) shall 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.

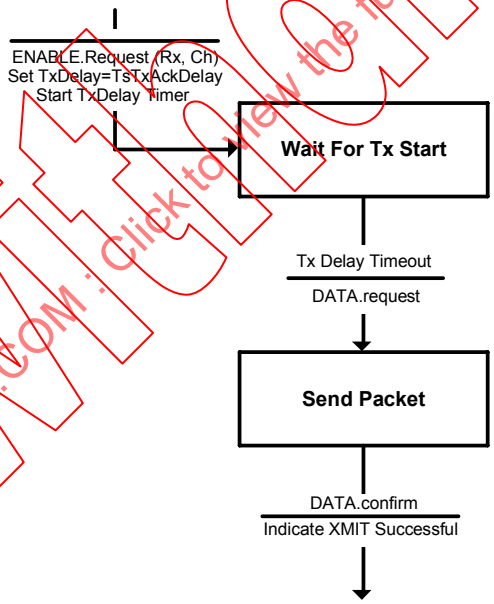


Figure 16 – ACK Transmission

5.5.5.3 XMIT

5.5.5.3.1 General

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.

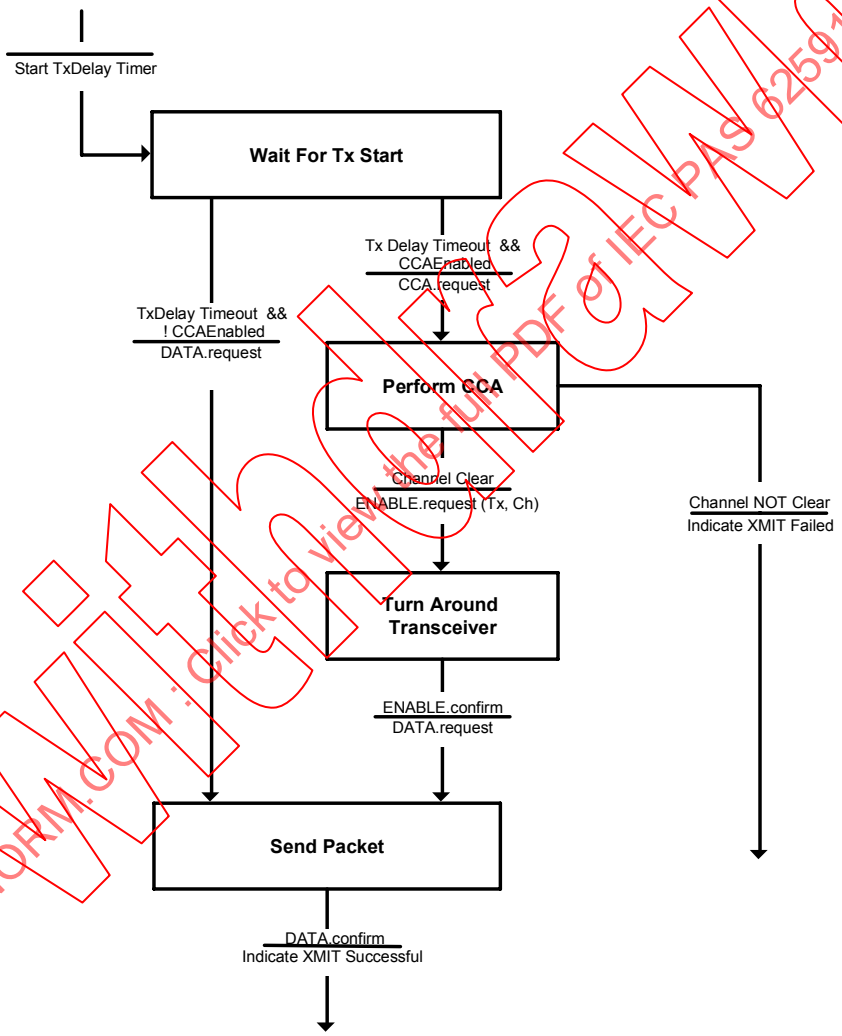


Figure 17 – Transmit State Machine

5.5.5.3.2 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.

5.5.5.3.3 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).

5.5.5.3.4 Turn-Around Transceiver

Upon confirming the channel is clear, the transceiver shall 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.

5.5.5.3.5 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.

5.5.5.4 RECV

5.5.5.4.1 General

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.

5.5.5.4.2 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.

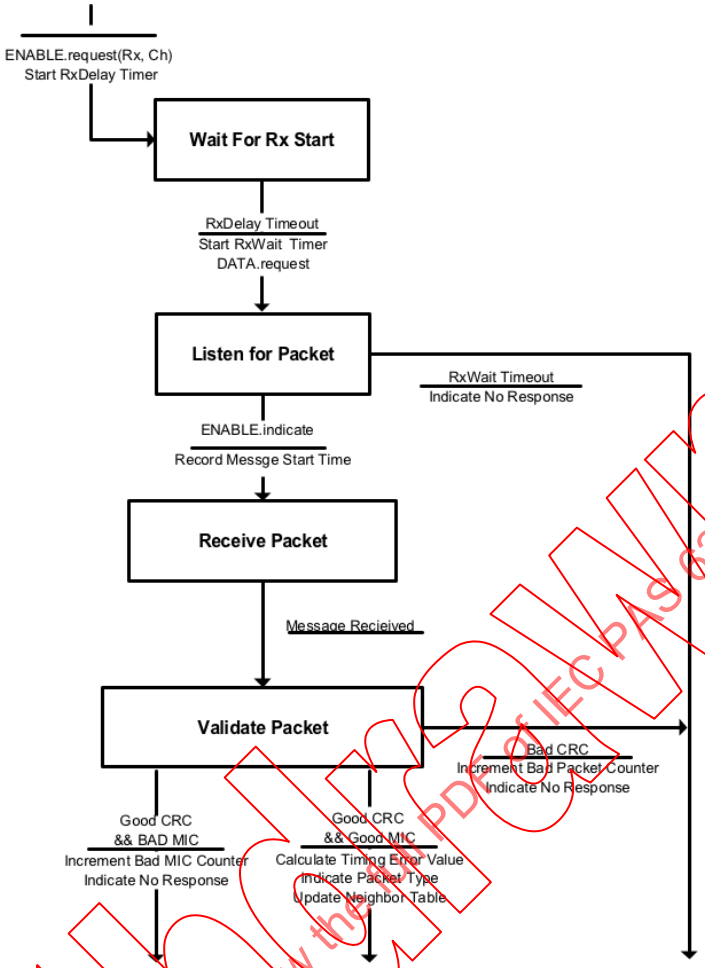


Figure 18 – Receive State Machine

5.5.5.4.3 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.

5.6 Physical Layer-Specific Requirements

Subclause 5.6 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, 2 450 MHz frequency channels to the indices used by the TDMA Data-Link Layer.

Table 12 – 2 450 MHz 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	2 120 µs ±100 µs
TsRxOffset	Start of the slot to when transceiver shall be listening	1 120 µs ±100 µs
TsRxWait	The time to wait for start of message	2 200 µs ±100 µs
TsMaxPacket	Maximum packet length (includes PhL header and DLPDU, i.e., 133 bytes)	4 256 µs
TsTxAckDelay	End of message to start of ACK	1 000 µs ±100 µs
TsRxAckDelay	End of message to when transceiver shall be listening for ACK	800 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	1 800 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	2 405
1	12	2 410
2	13	2 415
3	14	2 420
4	15	2 425
5	16	2 430
6	17	2 435
7	18	2 440
8	19	2 445
9	20	2 450
10	21	2 455
11	22	2 460
12	23	2 465
13	24	2 470
14	25	2 475
15		(Not Used)

The channel map array is two bytes long for IEEE STD 802.15.4-2006 (2 450 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).

6 Network Management

6.1 Purpose

Clause 6 specifies the Network Layer and Network Management requirements for WirelessHART Networks. Above the Network Layer resides the Application Layer that defines allowed data types, procedures and commands. Below the Network Layer resides the TDMA Data Link Layer. The rules necessary to communicate via the Network Layer over wireless HART Networks are specified in Clause 6.

Figure 19 shows the scope of this specification. Clause 6 provides the following:

- An overview of Network Layer requirements, and;
- Network Layer service requirements. The Network Layer service requirements are segregated into common communication services and technology specific management services.

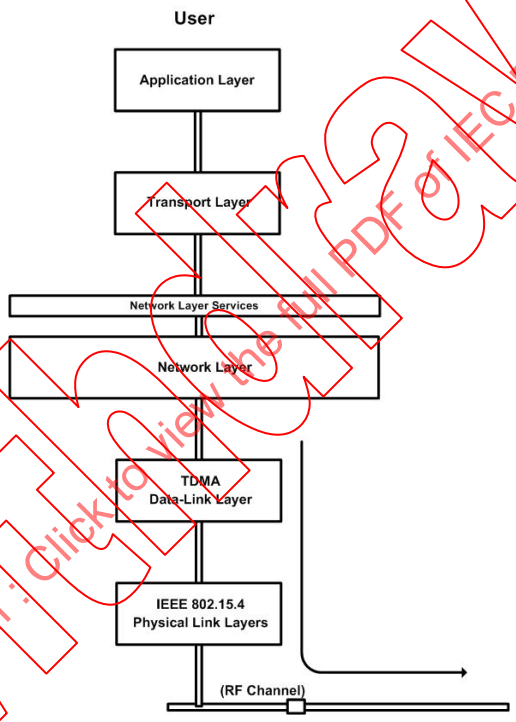


Figure 19 – Network Layer

The segregation of requirements into these categories is intended as a frame of reference rather than as a description of an actual implementation.

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

6.2 WirelessHART™

6.2.1 General

WirelessHART is a sensor mesh communication system (see Figure 20) that simplifies network and device installation and allows the end user to tailor the installation and its topology to satisfy specific application requirements. The basic elements of a WirelessHART network include:

- Field Devices that are connected to the Process or Plant Equipment. All network devices, including field devices, shall be able to source and sink packets and be capable of routing packets on behalf of other devices in the network.
- A Gateway that enables communication between Host Applications and WirelessHART field devices in the WirelessHART network. Every WirelessHART Network includes a WirelessHART Gateway. Gateways, in turn, may include one or more Access Points.
- A 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 shall be only one active network manager per WirelessHART network.

6.2.2 Mesh Networks

Figure 20 shows a basic WirelessHART network with the field devices deployed in a mesh topology. In this example network, there is 1 gateway, 1 network manager and 13 field devices (labeled a to n). All communication occurs, for example, by moving packets from the gateway, through the intermediate devices, to the packet's destination. Each movement of a packet from one device to another along the route to the packet's final destination is called a hop. Requirements specifying the communication of packets between adjacent network devices can be found in the TDMA Data-Link Layer Specification.

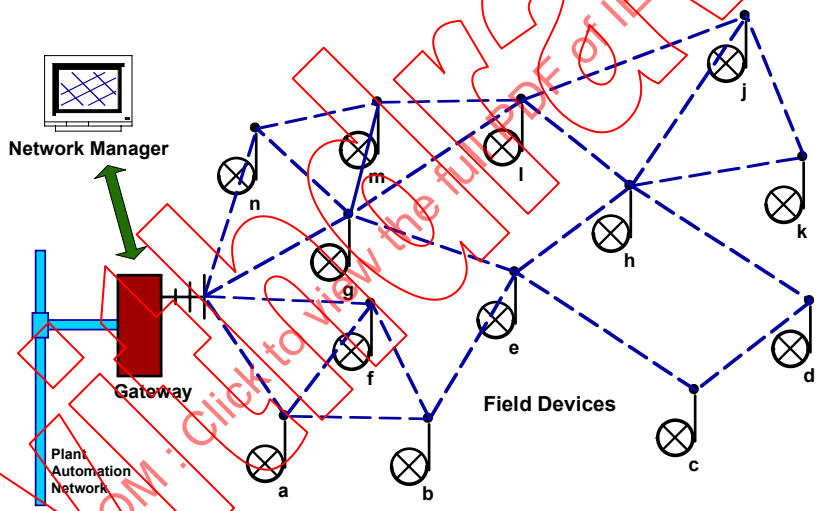


Figure 20 – WirelessHART Network

All devices shall be able to source and sink packets and be capable of routing packets on behalf of other devices in the network. 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.

Within this network, nodes a, f, g, and n are one hop from the gateway. Since these devices can pass packets directly to the gateway, communication with these devices has the lowest latency. However, since WirelessHART uses mesh technology, redundant links are included to improve system reliability by allowing packets to be routed around e.g. interference. In Figure 20, node a can communicate directly to the gateway but can also communicate via node f to the gateway for example if the direct route becomes blocked.

Nodes c, d, j and k are several hops away. All intermediate devices (e.g., field devices g and e) shall be capable of receiving and forwarding packets to and from these devices. Since the packets shall make several hops to and from nodes c, d, j and k, the communication latency to these devices is longer than the devices that are one hop away. However, WirelessHART mesh technology allows the physical size of the network to become larger and allows communication around obstructions. Furthermore the number of the devices that can be supported in a single network can be larger than that supported by star networks.

By supporting mesh communication technology, WirelessHART networks can be installed in a wide range of topologies. 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.

The WirelessHART Field Network maintains very high reliability using several mechanisms including multiple paths to network devices, multiple frequencies, and retries. If improved reliability is required, more paths can be inserted by adding additional access points and field devices. Additional network access points can be used to increase throughput and reduce latency.

Precise time synchronization is critical to the operation of networks based on time division multiplexing. Since all communication happens in slots, the Network Devices shall have the same notion of when each slot begins and ends, with minimal variation. In a typical WirelessHART Network, time propagates outwards from the Gateway.

6.2.3 WirelessHART Network Components

6.2.3.1 General

Subclause 6.2.3 discusses the types of devices and other elements that are associated with a WirelessHART installation. The WirelessHART Network supports a wide variety of devices from many manufactures. Figure 21 illustrates the basic elements of a WirelessHART installation along with types of equipment and functional components that may be present. In some cases, like a Field Device, the product may be a physical device. In other cases (e.g., Network Manager), a logical or abstract element is described. The items shown in Figure 21 include:

- Field devices are connected to, and characterize, or, control, the Process. They are both a producer and consumer of packets and shall be capable of routing packets on behalf of other Network Devices.
- Wireless Adapters connect existing wired field devices to the WirelessHART Network.
- A Gateway enables communications between Host Applications and devices that are members of the WirelessHART Network. The Gateway has one or more Access Points interconnecting the Plant Automation Network and the WirelessHART Network.
- Handhelds and other Maintenance Tools are portable applications used to configure, maintain or control plant assets. Only portable equipment directly connecting to the WirelessHART Network falls into this category.
- The 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 WirelessHART Network.
- The Plant Automation Network connects client applications to the gateway and, consequently, the WirelessHART Network's members.
- Host Applications are the tools used by plant staff to monitor, manage and control plant operations and plant equipment. Host Applications include all tools communicating to Network Devices via the Gateway (e.g., plant automation controllers and maintenance tools).

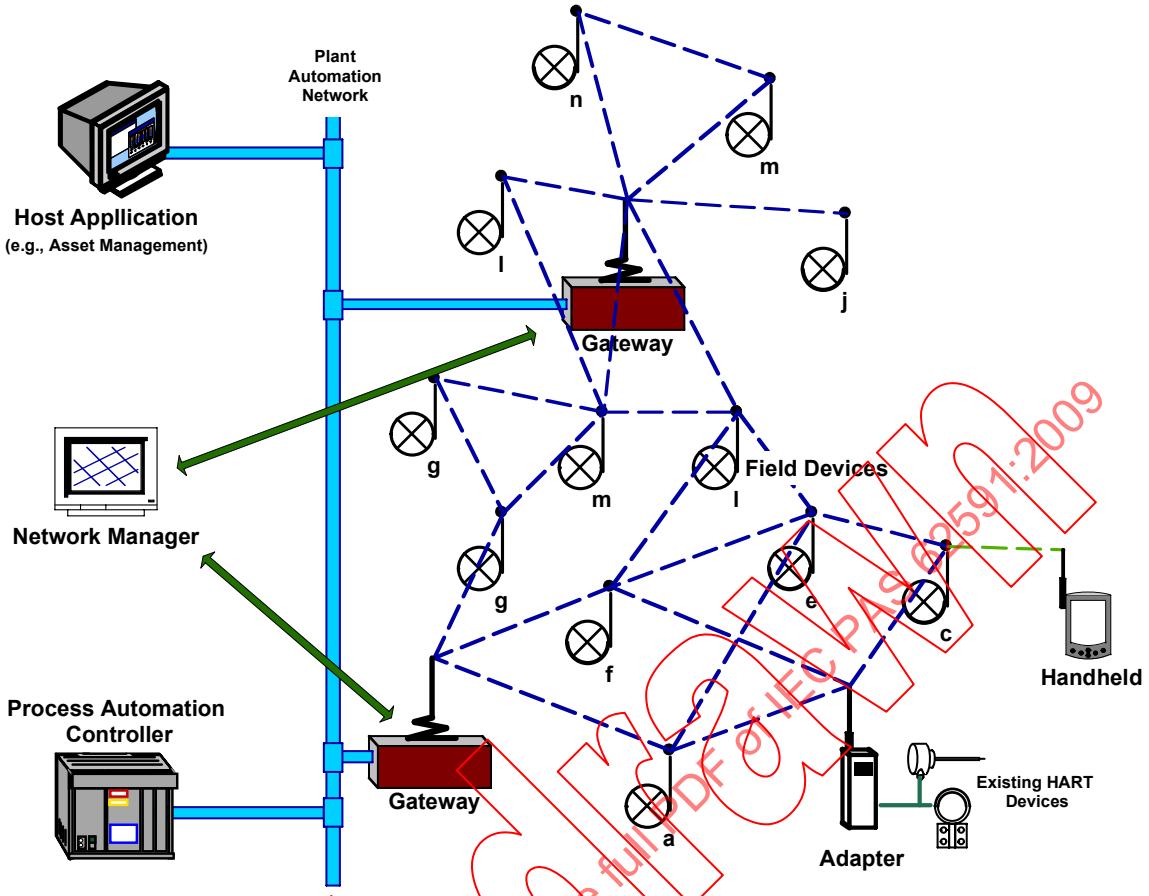


Figure 21 – Typical WirelessHART Network Components

6.2.3.2 Network Devices

Network Devices have a direct Physical Layer connection to the network. Each network device has a HART Unique Address that is used in communications with the device. Each Network Device also has properties holding information on update rates, sessions, and device resources covering items such as the size of the superframe, etc. Each Network Device contains a list of Neighbor Devices that it has identified during its listening operations (neighbors can be identified devices during any receive time slot).

Typical Network Devices include Field Devices, Wireless Adapters and Gateways. At least one of the Physical Layers supported by the Token-Passing Data-Link Layer shall be included in the Network Device.

Dedicated routers are also Network Devices. However, since all Network Devices shall be capable of routing, they can offer only a subset of the capabilities of, for example, a WirelessHART field device.

All Network Devices shall be capable of routing packets on behalf of other Network Devices. The Network Device uses internal routing tables to decide which Network Device to forward the packet to. If Graph Routing is used, then the Graph ID is used to select the neighbor to forward to. If Source Routing is used, then the next entry in the Source Route, or the final destination address itself, is used to determine the next Network Device to forward the packet to. Network Devices are described in detail in Wireless Device Specification.

6.2.3.3 WirelessHART Field Device

The most common type of Network Device is a Field Device. The WirelessHART field device is a Network Device that integrates wireless communications into the traditional HART field device. The field device may be line, loop, or battery powered or they may be powered in

some other fashion. The WirelessHART Field Device may or may not support the 4-20mA current loop signaling traditional process measurement and control devices.

6.2.3.4 Wireless Adapter

The Wireless Adapter is a Network Device that enables connection and communication with an existing field device in a WirelessHART Network. The Wireless Adapter connects the existing device to the WirelessHART Network and supports the publishing of process data and status on behalf of the connected device within the WirelessHART Network.

6.2.3.5 WirelessHART Gateway

A Gateway is a Network Device with one or more Access Points. The Gateway connects the WirelessHART Network to a plant automation network allowing data to flow between the two networks. Network Device data collected by the Gateway is communicated to the plant automation network using its protocols and interfaces. This communication includes, for example:

- Routine communication of process-related data and events. This communication is cyclical and occurs on a predictable periodic interval.
- Status and other event generated data communication occurs as the result of a field device maintenance or failure or as the result of abnormal process conditions. This communication is sporadic but shall occur in a timely fashion.
- Configuration and maintenance related communication generally occurs in bursts. Maintenance, configuration and diagnostic activities result in many packets over a short time interval being communicated to a specific field device. However, configuration and maintenance on a given device is infrequent (e.g., once in every three months or longer).

The Gateway connects host applications such as Process Automation Systems and Asset Management Systems to the WirelessHART Field Devices. A Gateway can be used to convert from one protocol to another, as go-between two or more networks that use the same protocol, or to convert commands and data from one format to another.

While a network has only one Gateway, in many situations the Gateway will be virtualized and supports more than one Access Point. These multiple Access Points each have their own physical address and are used to improve network throughput and reliability. In other words, more packets per second through the network are possible and the network is resistant to the failure of a single access point.

To simplify support for redundant Access Points, every Gateway has a fixed; well known address (Unique ID = 0xF981 0x000002; Nickname = 0xF981). Consequently, the Gateway resides at the root of all graphs allowing packets to be routed to the most convenient Access Point. Since all routes shall include alternate Access Points, should an Access Point fail then network traffic will be impaired but communication will still be possible. When a Gateway uses multiple Access Points the Network Manager shall provide redundant routing for each Network Device using at least two of the Gateway's Access Points.

The Gateway is the clock source for the network and one or more of its Access Points may propagate the clock to the network. If there are several Access Points providing the clock, it is the Gateway's responsibility to ensure they stay synchronized with each other. This leads to one of three possible configurations:

- a single Access Point (see Figure 22);
- multiple Access Points providing the network clock (see Figure 23); or,
- multiple Access Points with at least one not providing the network clock (see Figure 24).

In any case, there shall be at least one Access Point providing the clock to the network. When an Access Point does not source the network clock it shall synchronize to the Network Devices acting as its clock parents.

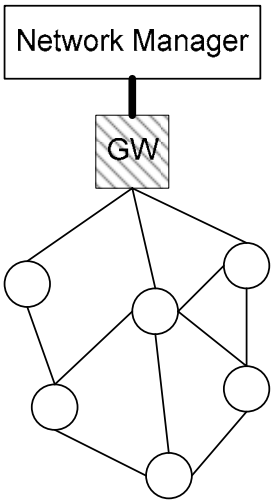


Figure 22 – Single Access Point with Clock

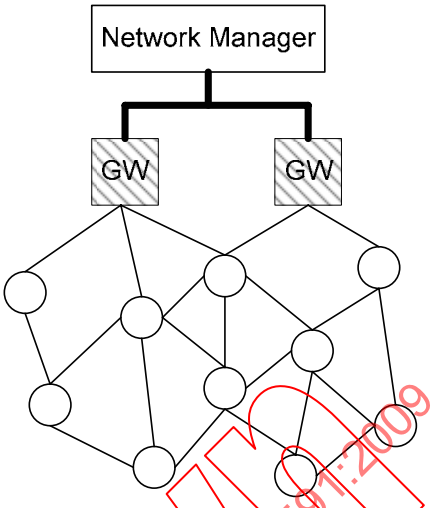


Figure 23 – Multiple Access Points with Clocks

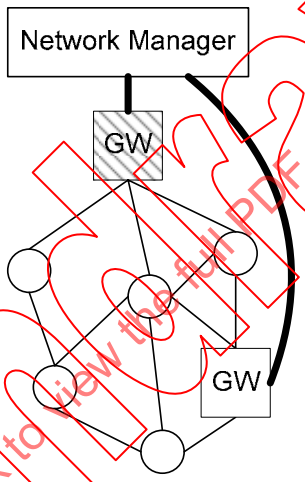


Figure 24 – Access Point Not Providing Clock

6.2.3.6 Network Manager

The Network Manager is an application that manages the WirelessHART Network and its Network Devices. The Network Manager forms the WirelessHART Network, joins and configures new Network Devices, and monitors the network.

The Network Manager contains a complete list of Network Devices and ensures each has a network-unique short 16-bit Nickname. The Network Device list maintained by the Network Manager is used for network functions such as routing and scheduling. The Network Manager is responsible for configuration of the network, scheduling communication between WirelessHART Devices, management of the routing tables and monitoring and reporting the health of the WirelessHART Network.

As part of its system functions, the Network Manager collects performance and diagnostic information. This information is accessible during run-time making it possible to view and analyze the behavior of the overall network. If problems are detected the reconfiguration of the network is performed while the network is operating. This network grooming is performed continuously as the overall network operation and performance varies due to changes in network load and environmental conditions.

While redundant Network Managers are possible, there shall be one and only one active Network Manager per WirelessHART Network. The Network Manager has a fixed, well known address (Unique ID = 0xF980 0x000001; Nickname = 0xF980).

Since the Network Manager is an application rather than a Network Device, the location of the Network Manager application is not restricted by this specification. However, the Network Manager shall have a secure communication channel to the Gateway and the Security Manager.

6.2.3.7 Security Manager

Join, Network and Session Keys shall be provided to the Network Manager and Join keys shall be provided to Network Devices. These keys are used for device authentication and encryption of data in the network. The Security Manager is responsible for the generation, storage, and management of keys. There is one Security Manager associated with each WirelessHART Network. The Security Manager may service multiple WirelessHART Networks.

While operation of, and requirements for the Security Manager are outside the scope of this specification, these applications are often used to manage these keys. The Security Manager may be a centralized function in some plant automation networks, servicing more than just one WirelessHART Network and in some cases other networks and applications.

6.2.3.8 WirelessHART Handheld

Portable computing is enabling the creation of the Wireless Worker. This worker represents the next generation instrument technician or plant operator. This nomadic worker is enabled using wireless technology (e.g., WiFi, WirelessHART, etc.) to electronically access process screens, plans, schematics, and other plant documentation while managing, maintaining, commissioning WirelessHART Devices.

The Handheld is a portable WirelessHART-enabled computer containing a Host Application. Handhelds are used to configure devices, run diagnostics, perform calibrations, and manage network information inside each device. Handhelds are not required to support routing. When used in a maintenance lab Handheld Devices can connect directly to WirelessHART Field devices through their FSK modem.

NOTE It is possible for Handhelds to access the Gateway directly via a Wi-Fi infrastructure. However, in this scenario the handheld is just another Host Application. In other words, in this specification, Handhelds are defined as connecting directly to the WirelessHART Network.

When operating with a formed WirelessHART Network, this Handheld joins to the target Network Device (i.e., the target device shall be within one-hop). When operating with a target Network Device that is not connected to a WirelessHART Network, the Handheld shall operate as the combination of a Gateway and Network Manager by forming its own WirelessHART Network with the target Network Device.

6.2.4 Message Routing

6.2.4.1 General

WirelessHART supports both Graph and Source routing of messages.

- Graph - The network topology can be represented as all of the devices and the directed links between them. A Graph Route is a subset of the directed links and devices that provides redundant communication routes between a source and a destination device. The route actually taken is based on current network conditions when the packet is conveyed from the source to the destination.
- Source - A Source Route is a single directed route (devices and links) between a source and a destination device. The source route is statically specified in the packet itself. Current network conditions could break the packet's specified route causing packet loss.

Only the ID of the Graph Route to be used is in the packet and, consequently devices in a Graph Route shall be configured prior to its use. Each intermediate Network Device is configured with a fragment of the overall Graph route. The device shall contain a list of all the

links that can be used to forward a packet along the Graph. Graph Routes are redundant, highly reliable, and, should be used for normal, routine communications (alarms, request/response, publishing, etc.) both upstream and downstream.

Source Routing contains the entire route specification in the pack and, consequently, intermediate devices require no knowledge of the Source Route in advance. As the packet is routed, each intermediate device propagates the packet to the next device in the packet's Source Route List. Source Routes are not redundant and may fail at anytime. Consequently, Source Routes should only be used for testing routes, troubleshooting network paths or for ad-hoc communications (e.g., routing Join Responses).

All devices shall support both Source and Graph routing.

6.2.4.2 Graph Routing

A Graph Route is a directed list of paths that connect network endpoints. The specific paths associated with each graph shall be explicitly configured by the Network Manager in the individual Network Devices. A single network instance may have multiple graphs, some of which may overlap. Each Network Device may have multiple graphs going through it, even to the same neighbors. Graphs are unidirectional.

Every graph in a network is associated with a unique Graph Id. When using Graph Routing, the device places the Graph ID value in network header. Devices receiving the packet then forward it along the set of paths belonging to the Graph, to the destination.

Each Graph ID in the device should have multiple associated neighbors. In a properly configured network, all devices will have at least two devices in the Graph through which they may send packets (ensuring redundancy and enhancing reliability). A device routing a packet shall lookup the Graph Id and then send the packet to any of the neighbors listed. Graph Routing is illustrated below in Figure 25.

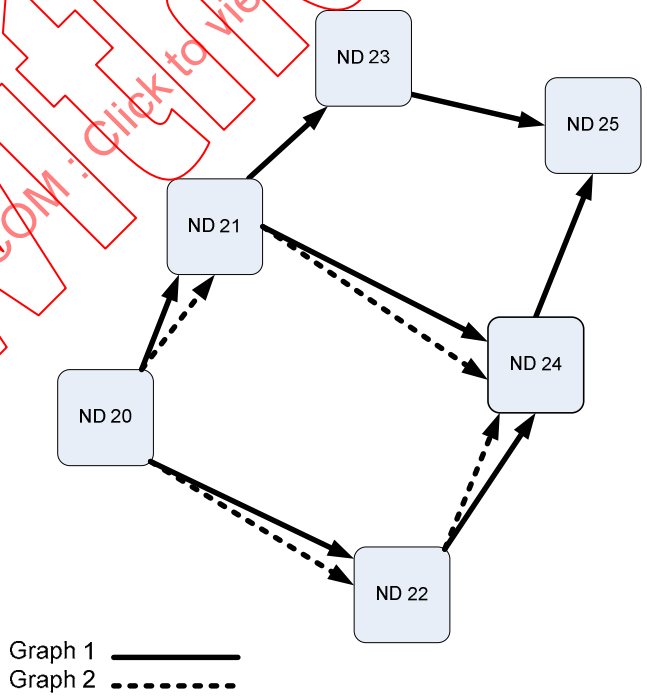


Figure 25 – Graph Routing

In Figure 25, ND 20 communicates with ND 25 using Graph 1. To send a packet on that graph, ND 20 may forward it to ND 21 or ND 22. From those devices, the packet may take several alternate routes, but either way, following Graph 1 it will end up at ND 25. Similarly, to communicate with ND 24, ND 20 sends packets on Graph 2 (i.e., through ND 21 or ND 22).

A special simplified version of Graph Routing, Superframe Routing, is also supported by all devices. When Superframe Routing is performed, the Superframe ID is placed in the NPDU. Devices receiving a Graph Routed NPDU attempt to lookup the Graph ID, and failing that, the Network Layer looks for a superframe with the same value. If successful, the device may forward the packet to any neighbor with a link in that superframe.

6.2.4.3 Source Routing

Source Routing specifies, in optional NPDU fields, the specific device-by-device route (i.e. a list of addresses) a packet shall take when traveling from the source device to the destination device. As the packet is propagated, each intermediate device looks at the source route address list and forwards the packet to the next device in the list. Source routing can only be used by Network Devices that have been configured with source routes' lists by the Network Manager because only the Network Manager knows the complete topology of the network. Source Routing is shown below in Figure 26.

Source Routing specifies a single path and, if one of the intermediate links fails, the packet is lost. Consequently, Source Routing is much less reliable than Graph Routing. When a source route fails, the device at the point of the failure shall notify the Network Manager. It is the responsibility of the Network Manager to take corrective action.

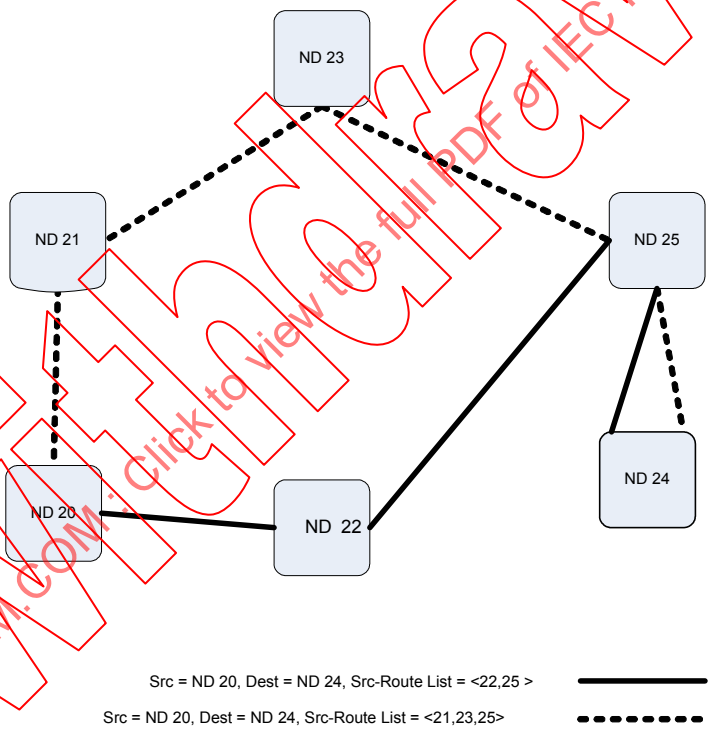


Figure 26 – Source Routing

In Figure 26, suppose that ND 20 wishes to send a packet to ND 24. The routing table on ND 20 may contain <21,23,25> as the source route for ND 24. In that case, ND 20 will originate a packet containing <21,23,25> in the header, and send the packet to ND 21. ND 21, upon receiving the packet will send it to ND 23 after finding it in the header. ND 23 will send the packet to ND 25. Finally, ND 25 will send the packet to ND 24 (the final destination). Alternatively, ND 20's route to ND 24 may be <22,25>. In that case, ND 20 will originate a packet containing <22,25> in the header, and send it to ND 22. In either case, the packet will end up at ND 24.

6.2.5 Security

End-to-end communications are managed on the Network Layer by sessions. A device may have more than one session defined for a given peer device. In fact, almost all network

devices will have at least two sessions with the Network Manager: one for pair-wise communication and one for network broadcast communication from the Network Manager. All devices will also have a Gateway session. The sessions are distinguished by the Network Device addresses assigned to them. For the pair-wise session with the Network Manager, a device's standard Network Device address will be used; for the broadcast session, the special Nickname address 0xFFFF will be used.

A network device shall keep track of security information (encryption keys, nonce counters) and transport information (reliable transport sequence numbers, retry counters, etc.) for each session in which it participates.

6.3 Network Layer Services

6.3.1 General

Subclause 6.3 specifies the operation of the Network Layer from a "black box" point of view. Subclause 6.3 specifies the Service Primitives (SPs) supplied by the Network Layer to the Application Layer. In addition to specifying the individual SPs, time sequence diagrams are included to indicate the order in which the SPs should be used and the order of event occurrence at the protocol layer boundaries.

The Services described in 6.3 are used to obtain:

- Message services supporting bi-directional request/response communication traffic and unidirectional notifications (e.g., for publishing process data).
- Management services for WirelessHART Network Layer configuration.

All SPs described here shall 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.

6.3.2 Network Layer Message SPs

6.3.2.1 General

Message SPs provide services supporting the basic transfer of data between devices. The Network Layer supports request/response communications and one-way notification traffic. In addition to normal request response traffic, the Application Layer may request guaranteed service. When requested, the guaranteed service will perform retries as needed to ensure a response is obtained from the destination device. The time sequence diagram for the message SPs is shown in Figure 27.

In Figure 27, Sequence 1 illustrates an un-acknowledged propagation of a data segment across the network.

The transmit sequences illustrates request/response traffic between devices. Sequence 2 shows a basic, acknowledged request/response transaction. In this sequence, the TRANSMIT.request inserts the message into the Network Layer's transmit queue. When a message is received and validated, the correspondent Network Layer generates a TRANSMIT.indicate. In return, the Application Layer generates a TRANSMIT.response, which is communicated to the source Network Layer. Upon reception of the response, the source Network Layer generates a TRANSMIT.confirm to the Application Layer.

Finally, in Sequence 3, the notification or publishing transaction is illustrated. In this transaction, the TRANSMIT.confirm is generated immediately after the message is conveyed

to the underlying Data-Link Layer. Publishing is an un-acknowledged and relies on over-sampled, repetitive operation to simplify the network transaction.

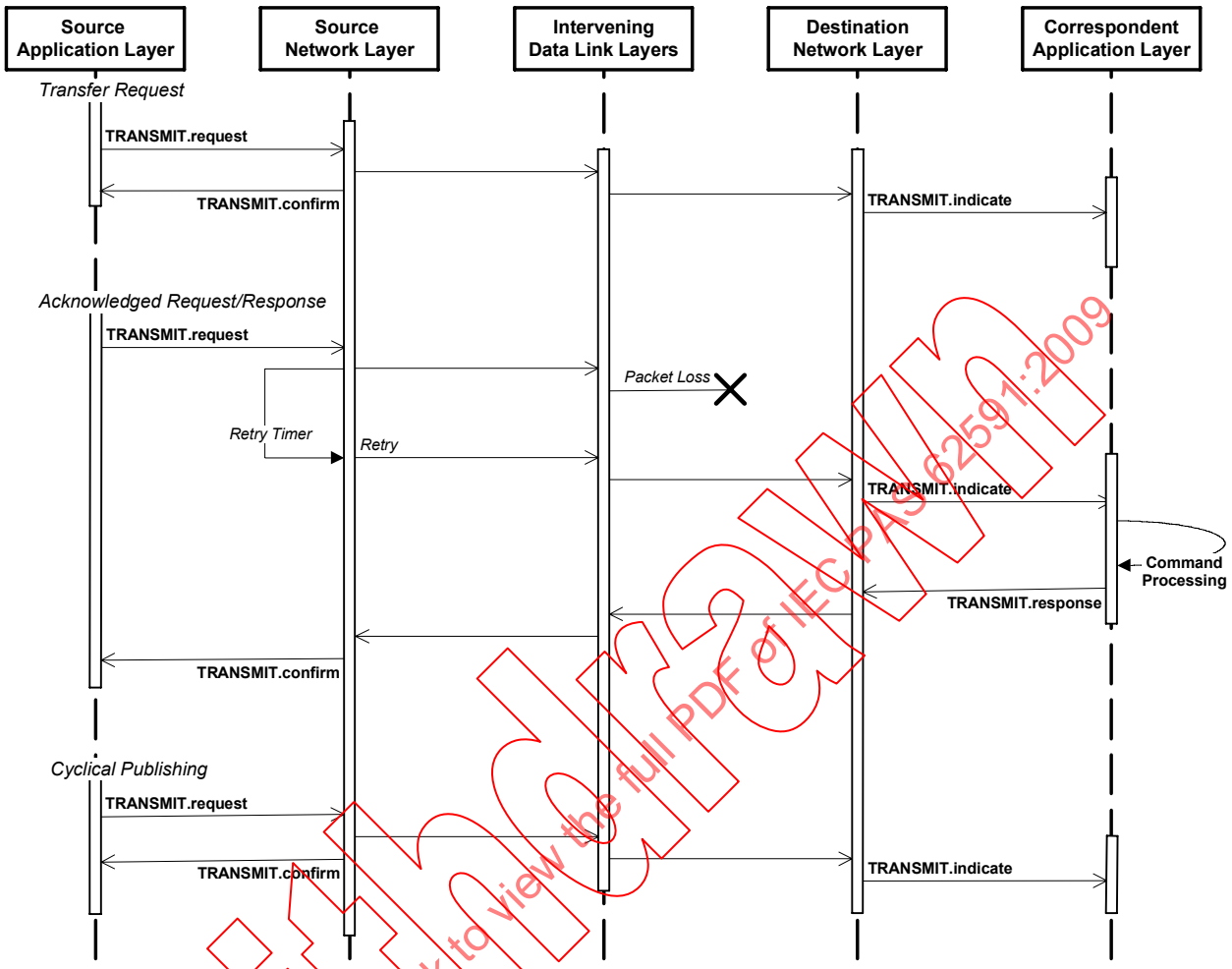


Figure 27 – Network Layer Message Sequence

6.3.2.2 TRANSMIT.request

TRANSMIT.request (handle, nickname, priority, transportType, payload)
TRANSMIT.request (handle, uniqueID, priority, transportType, payload)
TRANSMIT.request (handle, destEnum, priority, transportType, payload)

This overloaded SP is used by the Application Layer to send the packet to one or more devices in the network. The parameters included with the service request include:

- ***handle*** - The packetHandle is supported for the convenience of the Application Layer. The Network Layer returns this value in the corresponding TRANSMIT.confirm allowing the Application Layer to match requests with responses.
- ***uniqueID*** - the Unique ID (long address) of the destination device for payload. The NPDU shall include and the control byte shall indicate a long destination address.
- ***nickname*** - the Nickname (short address) of the destination device for payload. The NPDU shall include and the control byte shall indicate a short destination address.
- ***destEnum*** - The destination is enumerated and shall be one of the codes in Table 14. The NPDU shall include and the control byte shall indicate a short destination address.
- ***priority*** - The packet priority is determined by the contents of the payload and is one of {management, process data, normal, or alarm} see TDMA Data-Link Layer Specification for more information on packet priorities.

- ***transportType*** - The transport type (see Table 15) indicates the service requested and is used to set the Transport Byte (see Figure 35). The TRANSMIT.confirm is generated as follows:
  - For Transfer Request or Transfer Response service, the Network Layer promulgates the packet and immediately generates a TRANSMIT.confirm.
  - For a Request-Unicast, Request-Broadcast, or Search-Broadcast, the Network Layer shall perform retries to ensure a response from the destination device is received. If the retries are exhausted the TRANSMIT.confirm will indicate an error.
  - For Publish / Notify the Network Layer promulgates the packet and immediately generates a TRANSMIT.confirm. Publish transactions are typically used to cyclically publish process data.
- ***payload*** - The contents of payload parameter is transmitted to the destination device.

Table 14 – Destination Enumerator

Code	Destination
0	Network Manager
1	Gateway
2	Broadcast

Table 15 – Transport Type Codes

		Transport Byte Setting		
Code	**Description**	**B'cast**	**ACK'ed**	
0	**Transfer Request.** This is used by the Block Data Transfer Mechanism (master side)			➔ Req
1	**Transfer Response**. This is used by the Block Data Transfer Mechanism (slave side)			← Rsp
2	**Request-Unicast** (TRANSMIT.request only). This used by (master side) Devices executing Request/Response transactions (e.g., when configuring a device)		X	➔ Req
3	**Response-Unicast** (TRANSMIT.response only). This used by (slave side) Devices executing Request/Response transactions (e.g., when configuring a device)		X	← Rsp
4	**Search-Broadcast** (TRANSMIT.request only). This is used to send a broadcast message when attempting to identify a specific device (e.g., Command 21)	X		➔ Req
5	**Publish-Broadcast** (TRANSMIT.response only). This is a broadcast announcement to all network devices. (e.g., time broadcast)	X		← Rsp
6	**Request-Broadcast** (TRANSMIT.request only) (e.g., changing Network ID)	X	X	➔ Req
7	**Response-Broadcast** (TRANSMIT.response only). This is the unicast response to the corresponding Request-Broadcast	X	X	← Rsp
8	**Publish / Notify** (TRANSMIT.response only) (e.g., process data)			← Rsp

The Network Layer shall be capable of buffering at least one message in addition to the message associated with the current transaction.

When this SP is invoked, it shall validate the parameters (e.g., invalid destination address, no route to destination) and reject it if any errors are detected. In this case, the TRANSMIT.confirm SP shall be invoked indicating the error.

Otherwise the payload will the forwarded to the correspondent device by constructing the NPDU, invoking the Transport Layer, authenticating and enciphering NPDU and forwarding the NPDU to the Data-Link Layer.

6.3.2.3 TRANSMIT.indicate (handle, srcAddr, priority, transportType, payload)

This SP is invoked by the Network Layer when a packet is received and provides the payload to the client layer. Parameters included in the SP include the following:

- ***srcAddr*** - indicates the address of the source device generating the payload. This address depends on the network topology and may be, for example, the Primary Master, the Gateway, or the Network Manager.
- ***handle*** - The handle provided to the Network Layer. The handle shall be returned in the corresponding TRANSMIT.response SP.
- ***priority*** - The packet priority as provided in the TRANSMIT.request
- ***transportType*** - The transport type (see Table 15). The client layer shall respond based on the transportType as follows:
  - For a Transfer Request or Publish/Notify TRANSMIT.indicate no TRANSMIT.response shall be invoked. The transaction is complete.
  - For a Request-Unicast or Request-Broadcast the device shall generate a corresponding TRANSMIT.response SP.
  - For a Search-Broadcast the device may generate a corresponding TRANSMIT.response SP.
- ***payload*** - The data to being transported.

6.3.2.4 TRANSMIT.response (handle, payload)

This SP is executed by the Field Device to respond to all incoming TRANSMIT.indicate SP that require a response (e.g., transportType "Request-Unicast"). The handle shall be identical to that provided in the corresponding TRANSMIT.indicate SP.

The payload either contains the response data or a delayed response. All responses shall contain at least command completion status (i.e., the Response Code) for the command(s) in the TRANSMIT.indicate.

The transportType is inferred from the NPDU associated with the TRANSMIT.indicate and identifies the type of Network Layer transaction. Table 16 shows the transportType that shall be inferred based on the code provided in the TRANSMIT.indicate.

Table 16 – Transport Type Codes Pairs

TRANSMIT.indicate	TRANSMIT.response
Request-Unicast	Response-Unicast
Search-Broadcast Request-Broadcast	Response-Broadcast

The device uses the addresses and priority associated with the TRANSMIT.indicate (identified by the handle parameter) to generate and promulgate the NPDU back to the transaction originator.

6.3.2.5 TRANSMIT.confirm (handle, localStatus, [payload])

This SP is returned to the Application Layer to communicate the results of a previously executed TRANSMIT.request. The slave response (if any) is returned.

For request packets, TRANSMIT.confirm returns the response payload. For notification, it indicates the packet has been sent to the Data-Link Layer (and the payload is empty).

In all cases localStatus indicates the resulting status of the communication transaction. The localStatus indicates success, warning or error. For warnings and errors the localStatus shall have codes that indicate the cause of the warning or error (e.g., payload too large).

6.3.2.6 FLUSH.request (handle)

Deletes the indicated packet.

6.3.2.7 FLUSH.confirm (handle, localStatus)

Indicates whether the packet was deleted.

6.3.3 WirelessHART Network Layer Management Services

6.3.3.1 General

Management SPs support both configuration of the Network Layer and access to statistics that it gathers. The fundamental SP is a `LOCAL_MANAGEMENT` sequence.

NOTE None of the SPs in 6.3.3 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 configuration 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 Network 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.

6.3.3.2 LOCAL_MANAGEMENT.request(service, [data])

This SP is used to configure Network Layer properties. The parameters Services and Data are defined in Table 17 below.

6.3.3.3 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 the request.

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

This SP is used to notify `LOCAL_MANAGEMENT` of an un-requested Network Layer event report, see Table 17.

Table 17 – Local Device Management Commands

Service	Data	Description
RESET		Reset and initialize the Network Layer. All network tables are cleared when this primitive is invoked. This primitive is normally invoked on device power-up or when the device is being installed in a new network
WRITE_SESSION_KEY		Sets the session and nonce
	Unsigned-8 sessionId	
	Bits-8 sessionType	Bitmap {broadcast, unicast, join}
	Unsigned-40 destNodeAddress	Long Destination address (Unique ID)
	Unsigned-16 destNickname	Short Destination address
	Unsigned-16 myNickname	The device's Nickname or the Broadcast Address
	Unsigned-128 sessionKey	

Service	Data	Description
	Unsigned-32 correspondentNonceCounter	
	Unsigned-8 [1]numSessions	
DEL_SESSION		Delete a session
	Unsigned-8 sessionId	
	Unsigned-8 [1]numSessions	
ADD_ROUTE		Add route to a given destination address
	routeId	
	Unsigned-16 graphID	
	Bits-8 routeType	Bitmap: {Maintenance (default), Publish, Block Transfer, Event}
	Boolean isDefault	
	Boolean isGraphRoute	
	Unsigned-40 destUniqueID	Long Destination address
	Unsigned-16 destNickname	Short Destination address
	Unsigned-16 srcRouteHops [8]	Up to 8 Nicknames that lead to the destination. Unused addresses shall be set to the destination's Nickname
	Unsigned-8 [1]numRoutes	
DEL_ROUTE		Delete route information
	routeId	
	Unsigned-8 [1]numRoutes	
DEFAULT_ROUTE		Set given route as default
	routeId	
	Unsigned-8 [1]numRoutes	
READ_PDU_TIMEOUT		Reads the number of slots since packet's birth until it is discarded. Device checks ASN Snippet against current ASN
	Unsigned-16 maxPacketAge	Maximum number of slots a packet can live while hopping the mesh
WRITE_PDU_TIMEOUT		Writes the number of slots since packet's birth until it is discarded. Device checks ASN Snippet against current ASN
	Unsigned-16 maxPacketAge	
READ_TTL		The value TTL is initialized to when a new packet is generated
	Unsigned-8 TTL	
WRITE_TTL		
	Unsigned-8 TTL	

[1] Indicates that the value is returned.

6.3.3.5 Network Layer Constants and Attributes

Table 18, Table 19, and Table 20 show the attributes that are used for example in Clause 5.

Table 18 – General Network Layer Attributes

Attribute	Description
Unsigned-40 Unique ID	Used to construct the EUI-64 (long) address. (EUI-64 can be constructed by pre-pending the HCF OUI)
Unsigned-16 Nickname	Short Address
Unsigned-8 DefaultTTL	Packet life limit in hops. Specifies the number of hops a packet can travel before being discarded (defaults to 32)
Unsigned-16 maxPacketAge	Packet life limit in time using the ASN. Indicates the number slots after which a packet shall be discarded (defaults to 120 s - 12 000 slots)
Time HealthReportTime	Period at which to publish health reports (defaults to 15 min)
Time BcastReplyTime	Maximum amount of time to reply to a broadcast message (defaults to 60 s)
Time maxReplyTime	Used to trigger retires by Transport Layer (defaults to 30 s)
Unsigned-8 maxRetries	Number of retries used by the Transport Layer when performing an acknowledged packet transmission (defaults to 5).
Unsigned-8 minAdsNeeded	Preferred number of different Advertisements before issuing join request (defaults to 3).
Time AdWaitTimeout	The amount of time to wait while attempting to receive additional Advertisements (defaults to 300 s)
Unsigned-8 maxJoinRetries	Join retry limit (defaults to 5)
Time JoinRspTimeout	Join response timeout (defaults to Keep-Alive time)
Time ChannelSearchTime	The amount of time to stay on a given channel while listening for Advertise packets (defaults to 400 ms), see NOTE
Time ActiveSearchShedTime	Max amount of time to stay in active search mode while joining (defaults to 4 000 s). After this interval lapses the device transitions to passive search mode, see NOTE
Time PassiveCycleTime	When in passive search mode, the period over which the device cycles between sleeping and listening (defaults to 600 s). The sleep interval equals the PassiveCycleTime minus the PassiveWakeTime, see NOTE
Time PassiveWakeTime	When in passive search mode, the amount of time to be awake listening for the network (defaults to 6,5 s) , see NOTE
NOTE This actually is used by the TDMA Data-Link Layer, see Clause 5.	

Table 19 – Session Table Attributes

Attribute	Description
Unsigned-8 SessionTableSize	Size of session table (i.e., maximum number of entries)
Unsigned-8 NumSessionTableEntries	Number of entries currently in the SessionTable
SessionTable	Set of session table entries, see 6.4.2.4

Table 20 – Route Table Attributes

Attribute	Description
Unsigned-8 RouteTableSize	Size of route table (i.e., maximum number of entries)
Unsigned-8 NumRouteTableEntries	Number of entries currently in the RouteTable
RouteTable	Set of route table entries

6.4 WirelessHART Network Layer

6.4.1 General

The Network Layer provides routing, end-to-end security, and transport services. It manages "sessions" for end-to-end communication with correspondent devices. When packets are received via the Data-Link Layer's TRANSMIT.indicate SP, it transfers packets destined for the device itself from the Data-Link Layer to the client layer and routes packets destined for other devices by sending them back to the Data-Link Layer. It also processes packets received from the Application Layer with the TRANSMIT.request primitive.

The following diagram shows how data packets flow through the Network Layer.

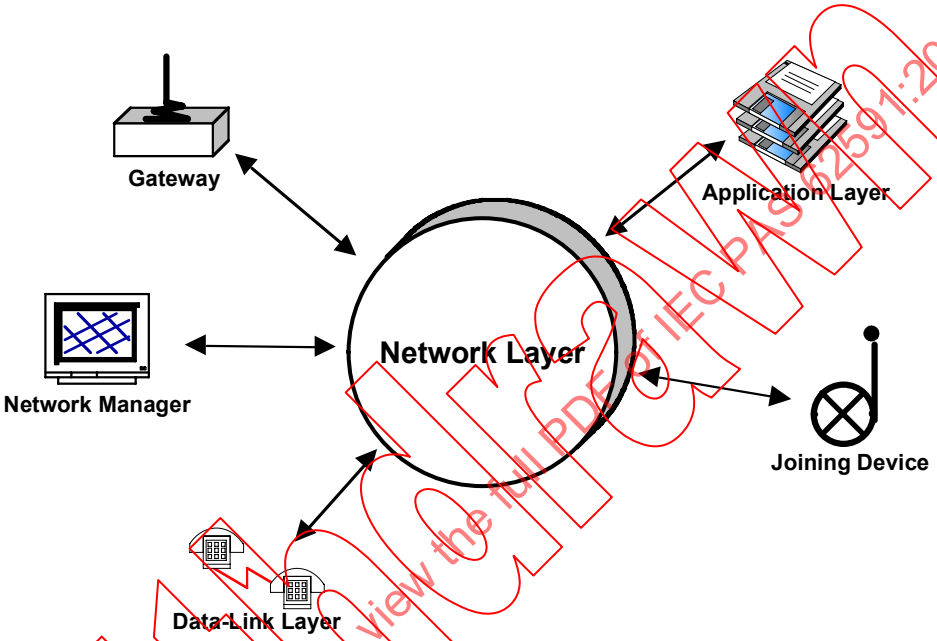
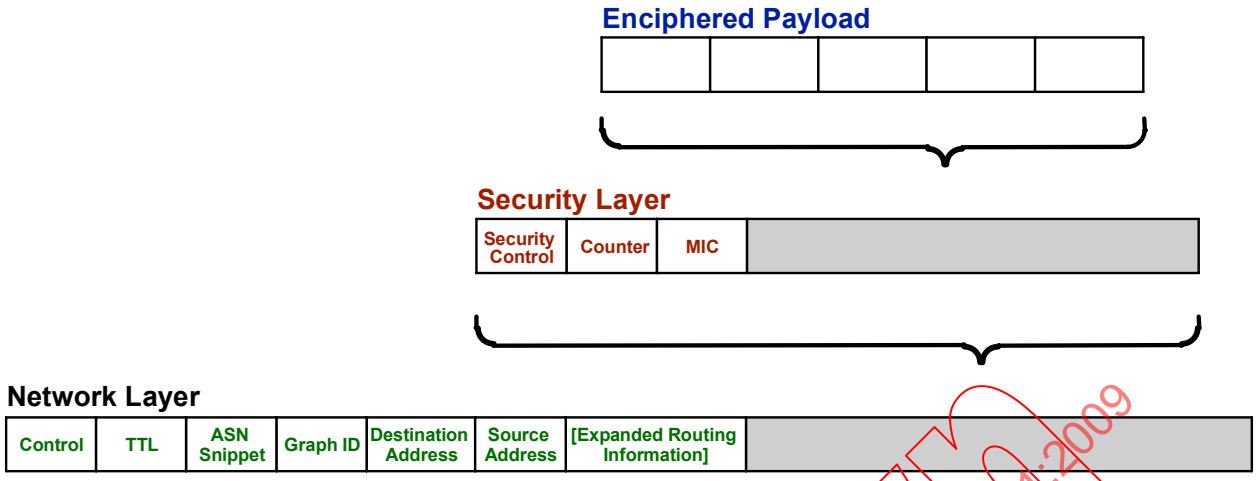


Figure 28 – WirelessHART Network Layer Context Diagram

6.4.2 Wireless Network Layer PDUs

6.4.2.1 General

As shown in Figure 29, the WirelessHART Network Layer PDU consists of three distinct functions. First the Network Layer fields consist of those fields required to route the NPDU to its final destination. On top of that is a layer of security fields used to ensure private, unmolested communication between the NPDU's end points. Finally, the NPDU payload is enciphered and contains the information being exchanged across the network.



NOTE Collectively these three elements comprise the NPDU.

Figure 29 – WirelessHART NPDU Structure

6.4.2.2 Network Layer

The Network Layer PDU segment consists of the following fields:

- A 1-byte Control field;
- The 1-byte Time To Live (TTL) hop counter;
- The least-significant two-bytes of the Absolute Slot Number (Latency Count);
- A 2-byte Graph ID;
- The (final) Destination and (original) Source Addresses; and
- Optional routing fields.

The complete Network Layer PDU consists of these fields plus the security fields followed by the enciphered NPDU payload.

6.4.2.2.1 Control Byte

The first byte in the Network PDU is the Control byte (see Figure 30). The first two bits (bit 7 and bit 6) indicate whether the source and destination addresses are long (8-byte) EUI-64 addresses or short (2-byte) Nicknames. The next three bits (bits 5-3) are reserved and no device shall make any assumption regarding its possible future use. Devices built before any such future use is assigned, shall set these bits to zero on transmission and masked off on reception.

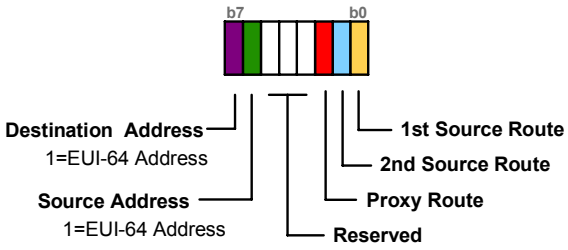


Figure 30 – Network Control Byte

Bits 2-0 indicate, when set, the presence of the optional routing fields. When present, the proxy route is a 2-byte field and each Source Route is 8-bytes long. Consequently, based on bits 2-0, the length of the header can be extended by 0, 2, 10, or 18 bytes.

When the Network Layer receives a NPDU, the Destination Address is inspected and, if the address matches the device's, then, the NPDU is authenticated and the payload deciphered. Once successful, the TRANSMIT.indicate primitive is invoked with the payload.

6.4.2.2.2 Time-To-Live

If the destination address does not match the device's, then the TTL counter shall be evaluated to determine whether the Network Layer discards or forwards the packet. The TTL counter controls the Time-To-Live for the packet and shall be decremented on each hop the packet takes toward its final destination. When TTL reaches 0, the packet shall not be forwarded to another device. If, when the packet is received by the Network Layer, the TTL is 0xFF, then the TTL in not decremented and the packet is always forwarded onward toward its final destination (i.e., TTL is infinite).

6.4.2.2.3 ASN Snippet

The ASN Snippet field is set to the least significant 16 bits of the Absolute Slot Number when the Network Layer's TRANSMIT.request SP is invoked. This field provides coarse but critical real-time performance metrics and diagnostic information on the operation of the network. It also provides, when the full ASN is recreated, the age of the packet.

If the TTL is valid, then the ASN corresponding to the packet's birthday is compared to the current ASN to get the age of the packet. If the age is greater than the maxPacketAge, the packet is discarded. The result of this comparison shall be provided as the timeout to the Data-Link.

6.4.2.2.4 Graph ID

The Graph ID is used to route the packet to its final destination. The Graph ID identifies a list of nodes, any of which can be used for forwarding the packet toward the final destination.

Otherwise, the packet is forwarded based on the Graph ID and the other routing information (if present) included in the Network Layer Header.

6.4.2.2.5 Source/Destination Addresses

The Source and Destination Addresses are each either 2 or 8 byte addresses. For more information, see the TDMA Data-Link Layer. These addresses are not modified during propagation of the NPDU.

6.4.2.3 Special Routes

6.4.2.3.1 General

If any of bits 2-0 are set, then one or more of the optional routing fields shown in Figure 31 are present in the NPDU header. When multiple fields are present, they are included in the NPDU header in the order depicted. These fields include the following:

- An optional 2-byte Nickname address for the proxy device; or
- Up to two optional Source Route segments each containing four Nicknames (8-bytes each).

Figure 31 – Expanded Routing Information

6.4.2.3.2 Proxy Route

Proxy routing is used to communicate with devices that have not yet been integrated into the network by the Network Manager. These include devices that are quarantined or are in the process of joining. When bit 2 of the NPDU Control byte is set, the 2 byte Nickname of the proxy parent for the final destination is indicated in the Proxy Address.

If the Proxy Address matches, the device is responsible for forwarding the NPDU to the device indicated by the Destination Address (see Figure 29 above). When proxy routing, the destination address shall be a EUI-64 address.

6.4.2.3.3 Source Route

If the source route field is present, the addresses it contains are used to route the packet to its final destination. Source routes should only be used to test or troubleshoot network paths.

Each Source Route field contains four addresses that designate the route the packet shall follow from the original source device to the packet's final destination. Each address is a 2-byte Nickname. Each byte of the addresses not used shall be set to 0xFF.

6.4.2.4 Security Sub-Layer

6.4.2.4.1 General

The security layer header of the NPDU (see Figure 32) is designed to ensure private, unmolested communication and to allow for future possible enhancements to WirelessHART security. To this end, the security layer starts with a Security Control Byte that specifies the security employed. This field is followed by the fields needed by the security algorithms employed.

Figure 32 – Security Sub-Layer

As indicated in Figure 33, the Security Control byte consists of a 4 bit enumeration (bits 0-3) that indicates the security strategy employed for this NPDU. The most significant bits (bits 4-7) of the Security Control byte are reserved and no device shall make any assumption regarding their possible future use. Implementations shall mask off the most significant 4 bits. Devices built before any such future use is assigned shall set these bits to zero on transmission.

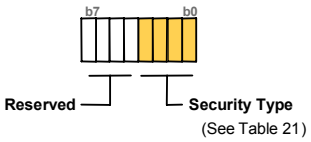


Figure 33 – Security Control Byte

The overall length of the security layer header depends on the type of security employed (see Table 21). Furthermore, the NPDU payload is always encrypted using the technique indicated by the Security Type Sub-field.

Table 21 – Security Layer Sizes

Security Type	Security Type Code	Counter Length	MIC	Total Length
Session Keyed	0	8 bit (LSB of 32bit Nonce Counter)	32 bits	6 bytes
Join Keyed	1	32 bit Nonce Counter	32 bits	9 bytes
Handheld Keyed	2	32 bit Nonce Counter	32 bits	9 bytes

6.4.2.4.2 Sessions

For security, WirelessHART is session oriented and all devices shall support multiple sessions. A session enables private and secure communication between a pair of network addresses. Four sessions are generally set up as soon as the device joins the network:

1. Network Manager and the device (unicast). This session is used by the Network Manager to manage the device.
2. Network Manager broadcast (to all devices in the network). All devices in the network have the same key for Network Manager broadcasts. This session is used to globally manage devices. For example, this can be used to roll a new Network Key out to the network.
3. Gateway and device (unicast). This carries normal communications (e.g., process data) between the gateway and the device.
4. Gateway broadcast (to all devices in the network).

Additional sessions may be added (e.g., to a handheld). Other than the Join session, the Network Manager creates all sessions and their corresponding key (e.g., when the device joins the network). Only the Network Manager may create or modify sessions. The Join key is unique, it is the only key that can be written by the Network Manager or using the device's maintenance port. Join sessions are always present in the device and cannot be deleted.

It is possible to deploy a session that connects two arbitrary devices in the network. However, this can result in undetected and unmonitored communication between the devices and, consequently, represents a security and safety threat. If the Network Manager supports peer-to-peer sessions between field devices all of the resulting communications should be routed via the Gateway thus allowing the detection and disruption of malicious behavior.

When and only when a new session is created, myNonceCounter is reset to zero. Each element in the peerNonceCounter list is set to the Nonce Counter Value in the Write Session command. The Session Key itself is a write-only value. No network device shall provide a means to read back any key (Session, Join, Network).

The attributes of a session are shown in Table 22. The combination of the correspondent address and session type shall be unique (no other session may have both the same correspondent address and session type entries).

Table 22 – Session Table Entry

Name	Description
Enum-2 sessionType	One of {Unicast, Broadcast, Join}
Unsigned-40 peerUniqueID	Long address of correspondent (Expanded Device Type Code + Device ID). See NOTE
Unsigned-16 peerNickname	Short address of correspondent device
Unsigned-128 sessionKey	(Write Only) Session key
Unsigned-32 peerNonceCounter	Largest nonce counter value received from the correspondent device
Bits-32 nonceCounterHistory	An array of bits recording the nonce counters received. Most significant bit is always set and corresponds to the current peerNonceCounter value. The least significant bit corresponds to 1+peerNonceCounter - sizeof (nonceCounterHistory) This creates a sliding window with each set bit recording a NPDU received with the corresponding nonce counter. The size of (nonceCounterHistory) equals the number of received NPDUs recorded. As the nonceCounterHistory is right shifted the history of the older NPDU are dropped
Unsigned-32 myNonceCounter	Nonce counter for packets sourced by the device
Ref transportable	Reference to Transport tables (normally 2) associated with correspondent
Ref routeTable	Reference to Route tables associated with correspondent destination
NOTE The EUI-64 is constructed by pre-pending the HCF OUI.	

6.4.2.4.3 Join Sessions

Join sessions are unique and are created by the field device when the join key is first written via the maintenance port. The addresses of the Join session are set to the standard Network Manager addresses. The myNonceCounter entry is set to zero when the session is created. The field device's myNonceCounter is used for both join requests from the field device and join responses from the Network Manager. To protect against replay attacks on the Join key myNonceCounter is non-volatile.

Several strategies might be used to create the Join session on the Network Manager. For example, a Join session for a specific field device could be created in advance using the field device's Unique ID and the Join Key. No matter the strategy used to create the join session, when the Network Manager receives and validates a join request the NPDU counter field is copied to the session's nonce counter entries and is used to construct the join response.

6.4.2.4.4 NPDU Encipherment

A four-byte, keyed Message Integrity Code (MIC) is used for authentication of NPDUs and the deciphering of the Network Layer payload. All NPDUs that fail to authenticate shall be discarded. The NPDU Header is not enciphered to allow intermediate devices to successfully route the packet. The NPDU payload is enciphered to ensure communications remain private and secure.

A keyed MIC is used to ensure that the NPDU arrives successfully and unmolested from the indicated source device. 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.

The NPDU payload is enciphered and is the byte-string 'm'. The NPDU header, from the NPDU Control byte through the NPDU MIC, is the byte-string 'a'. The TTL, Counter and MIC fields in byte-string 'a' are set to zero while enciphering the NPDU. These are replaced with their actual values before transmitting the packet.

The Network Layer Nonce (the 'N' byte-string) is 13-bytes long and shown in Table 23. Except for join responses, to create the Nonce:

- N[0] is set to zero.
- myNonceCounter is pre-incremented by one and written to the Nonce.
- The NPDU source address field is loaded into the Nonce (either the EUI-64 address or the zero-padded Nickname).

For join responses, create the Nonce by:

- Setting N[0] to one;
- Using the peerNonceCounter value (from the join request) as the Nonce Counter; and
- Loading the joining device's EUI-64 address into the Nonce.

Table 23 – NPDU Nonce (Byte-String 'N')

Byte	Format	Description
0	Unsigned-8	Set to 1 if Join Response otherwise set to 0
1-4	Unsigned-32	Nonce Counter (starting with MSB in N[1])
For EUI-64 source address:		
5	Unsigned-8	0x00 (HCF OUI)
6	Unsigned-8	0x1B
7	Unsigned-8	0x1E
8-9	Unsigned-16	Expanded Device Type Code (starting with MSB in N[8])
10-12	Unsigned-24	Device ID (starting with MSB in N[10])
For Nickname source address:		
5-10	Unsigned-48	Each byte set to 0x00
11-12	Unsigned-16	Nickname (starting with MSB in N[11])

Once the Nonce is constructed, the NPDU is enciphered. Once encipherment is complete, the TTL field is initialized and the NPDU security fields are populated. The entire NPDU is assembled by concatenating the NPDU Header, the security header, and the enciphered NPDU payload. Next, the NPDU is passed to the Data-Link layer for propagation.

6.4.2.4.5 NPDU Authentication

6.4.2.4.5.1 General

At the destination device, the NPDU is once again processed using the AES-128 engine to authenticate the NPDU and decipher the payload. This cipher requires four byte-strings as parameters:

- 'a', is the NPDU header (NPDU Control through MIC) with the NPDU TTL, Counter and MIC fields set to zero;
- 'm', is the NPDU payload;
- 'N', the 13-byte nonce (see Table 11); and
- 'K', the 128-bit AES Key.

Authentication and decipherment start by locating the correct session to determine the key and nonce counters. Next, the Nonce is constructed.

6.4.2.4.5.2 Nonce Re-Construction

The NPDU Nonce is re-constructed from the source address and by re-constructing (if necessary) the nonce counter. First, if the NPDU is a join response, N[0] is set to one otherwise it is set to zero. The NPDU source address field is loaded into the Nonce (either the EUI-64 address or the zero-padded Nickname). Next, the nonce counter is constructed:

- If the message is a join request, then the NPDU Counter is four-bytes long and shall be copied to the nonce counter.
- Else, if the message is a join response then the NPDU Counter is compared to myNonceCounter. If they do not match, the packet shall be discarded. If they match, the myNonceCounter is copied to the nonce counter (N[1] - N[4]).
- Else, the NPDU Counter is one-byte long and the nonce counter shall be reconstructed. To do this, the most significant three bytes of the peerNonceCounter are copied to N[1] - N[3]. If the NPDU Counter value is less than the quantity (1+LSB(peerNonceCounter) - sizeof(nonceCounterHistory)) then the 24-bit value in N[1]-N[3] is incremented. The NPDU Counter is copied to N[4].

The resulting nonce counter is compared to the nonceCounterHistory. If it corresponds to any of the bits set in the nonceCounterHistory, the packet shall be discarded.

6.4.2.4.5.3 Authentication

If the nonce is successfully constructed and the packet verified to be unique (i.e. it is not a replay of a previous packet), authentication and decipherment can be performed. If the authentication fails, the packet is discarded.

6.4.2.4.5.4 Replay Protection

If the packet is authentic, then the nonceCounterHistory is updated. Since NPDUs can arrive out of order or be lost completely, this sliding window algorithm is used to facilitate the communications and eliminate duplicate packets. The peerNonceCounter plus the nonceCounterHistory chronicles the NPDUs sinked by the device.

NOTE If the PDU has a nonce counter less than can be recorded in the nonceCounterHistory, the PDU will be discarded.

When a new Nonce Counter (N[1] - N[4]) value is encountered, the peerNonceCounter plus the nonceCounterHistory shall be updated to record its reception. If the received Nonce Counter is less than the peerNonceCounter, the appropriate bit in the nonceCounterHistory shall be set.

Otherwise, the nonceCounterHistory shall be right shifted and the peerNonceCounter value updated. This is accomplished by subtracting the peerNonceCounter from the Nonce Counter. The resulting difference indicates the number of times the nonceCounterHistory shall be right-shifted. After shifting, the MS bit of the nonceCounterHistory shall be set. Finally, the Nonce Counter is copied to the peerNonceCounter in the session table entry.

6.4.2.5 Payload

For security, the payload field is always enciphered to prevent observation by intermediate devices as the NPDU traverses the network. The payload consists of Transport Layer information, Transaction ID, Device Status, Extended Field Device Status, and one or more commands.

6.4.3 Wireless Transport Layer

6.4.3.1 General

The Data-Link Layer ensures packets are successfully propagated from one device to another. The Transport Layer can be used to ensure end to end communication is successful. In other words, the Transport Layer can ensure packets are communicated successfully across multiple hops to their final destination. The Transport Layer supports both acknowledged and un-acknowledged transactions.

Un-acknowledged service allows devices to send packets without requiring end to end acknowledgement and with no guarantee of packet ordering at the destination device. This method is useful, for example, for publishing process data. Since process data is propagated periodically, end to end acknowledgement and retries have limited utility considering a new data point will be generated on a regular basis

In contrast, the acknowledged service is used to construct a synchronous transport pipe across the network connecting to devices. The "transport pipe" allows devices to send packets and confirm their delivery. The Transport Layer orders the packets sent between devices and tracks their delivery. This method is best suited for request/response traffic and event notifications. When the acknowledged service is employed, the communication is synchronous. Only one transaction across the bus for that transport pipe is performed at a time.

Using acknowledged service allows the client layer to ensure retries are automatically performed if communication latency becomes excessive. Statistically, communication reliability for a well-formed WirelessHART mesh network is greater than 3 σ (3-sigma = 99,73 0020 4 %) and normally greater than 4 σ (99,99 95 %). Consequently, the primary benefit provided by the acknowledged service is that the client layer will receive positive acknowledgement after NPDU arrival at the destination. This allows synchronous operation across the network as well as assuring the client layer that the NPDU was delivered and acknowledged.

6.4.3.2 Transport Layer PDU

6.4.3.2.1 General

Subclause 6.4.3 specifies the format of the Transport Layer packet (TPDU). Each TPDU consists of the following fields:

- A transport byte used to ensure end to end packet delivery;
- The Device Status and Extended Device Status bytes; and
- One or more HART commands.

Figure 34 below illustrates the basic TPDU structure.

When the Network Layer TRANSMIT.request SP is invoked, the transportType is indicated. This allows the Transport Layer to correctly configure the Transport byte (see Figure 35 hereinafter). A Transport Layer transaction is modeled as follows:

- A "master" issuing a request packet and one or more "slaves" replying with a response packet; or
- A slave publishing a response packet.

For request/response transactions, the master may generate a broadcast request to the entire network. For example, the Gateway uses broadcast to "Poll by Long Tag" to find the device with the desired name.

Alternatively, a master may generate a packet and propagate the request to a specific device (i.e., unicast request). For example, a path down alarm generated by a field device is a request directed to the Network Manager. In turn, the Network Manager shall generate a response packet and send it to the device acknowledging receipt of the alarm notification.

For transactions consisting of only a response the Transport Layer simply pushes the packet to the destination. Publish Data Message operations use this technique.

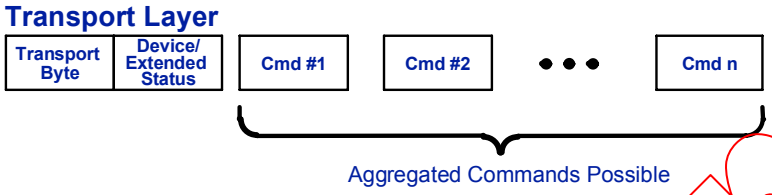


Figure 34 – Transport Layer

6.4.3.2.2 Transport Byte

The first byte in the TPDU is the transport byte (see Figure 35). It contains 4 subfields:

1. The most significant bit (bit 7) is set when a transport pipe (i.e., acknowledged service) is to be used. When set, the Transport Layer ensures that all requests are acknowledged and generate the TRANSMIT.confirm SP upon reception of the acknowledgement.
2. The next bit (bit 6) is set when the NPDU is a response packet. If the response bit is set, the payload shall only contain command responses.

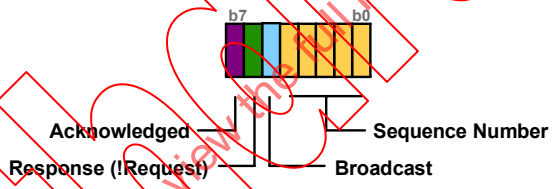


Figure 35 – Transport Byte

3. Bit 5, when set, indicates that a broadcast transport pipe is being used. This bit is set identically in the device response NPDU as in the request NPDU. This bit is used to identify the session (unicast or broadcast) that is the parent of the transport pipe and thus locate the correct transport table entry. The broadcast request may target all devices or (as indicated in a payload containing a Command 21 request) a single device. All devices receiving a broadcast NPDU that applies to themselves shall generate a response packet directed to the source device.
4. The sequence number field used to order and track packet traffic.
   - For acknowledged traffic, the sequence number is incremented by the transport master when it generates a request.
   - For un-acknowledged communications the sequence number shall be set to the least significant 5 bits of the packetHandle provided by the client layer.
   - In response packets, the slave returns the sequence number found in the request.

6.4.3.2.3 Device/Extended Status

The Device Status and Extended Device Status bytes are included in all TPDUs. Information about the Extended Device Status byte can be found in HCF Enumeration Table 17. (See HART Communication Foundation Common Table Specification HCF_SPEC-183, available at <http://www.hartcomm2.org/hcf/services_tools/doc_sales.html>

6.4.3.2.4 Aggregated Commands

With some limitations, WirelessHART allows multiple HART commands to be transported in a single transaction. This is especially useful when reading device configurations. WirelessHART natively supports HART 16-bit command numbers. The format of commands transported over WirelessHART is shown in Figure 36. They consist of a 16-bit command number, the command's length (1-byte) and the data field. Commands 0-255 are zero filled to form the 16-bit command number (e.g., Command 9 is transmitted as 0x0009). Command 31 is reserved and any response to a Command 31 request shall respond with "Command Not Implemented".

Figure 36 – WirelessHART Command Format

The transport byte indicates whether the transaction is a request or a response. For responses, the first data byte is always the Response Code. If the data field is too long to fit in the NPDU, then "Payload Too Long" (Response Code 60) shall be returned.

General requirements for aggregating commands include the following:

- In general, WirelessHART commands (0x0300 - 0x03FF) are not be aggregated with other HART commands. This simplifies command parsing in multi-processor designs.
- Only network manager write commands shall be included in "Network Management" transactions. Network Management transactions are acknowledged NPDUs containing write commands that modify network behavior.
- Many HART commands are fully autonomous and some host applications can "fire and forget" these commands. For example, in many cases, a combination of read commands can be aggregated in "Routine" transactions.

Additional restrictions may be imposed in the requirements for specific commands. Furthermore, certain processes (e.g., device calibration) may prevent command aggregation. This behavior is already supported in host applications.

6.4.3.3 Transport Table

Like security sessions, devices shall track multiple transport pipes. The properties that shall be managed for each active transport session are shown in Table 5 above. For each acknowledged communication link a new entry in the Transport Table entry is created. Field devices will act as a slave in at least three cases (Unicast with Network Manager, Broadcast with Network Manager, and Unicast with Gateway). These support request/response traffic used to configure and manage the field device. In addition, the device will act as a master during event notifications to the Network Manager and the Gateway (adding two more transport sessions).

NOTE Data Publishing, Notifications and Block Data Transfer are unacknowledged and, consequently, do not require a transport table entry.

On the other hand the Gateway and the Network Manager will need to track many transport sessions using several for each device. For example, when an acknowledged broadcast is generated by the Network Manager or Gateway, the Transport Layer shall ensure acknowledges are received from all affected devices (e.g., every device in the network). In this case the transport table tracks considerably more information than shown in Table 24.

Each entry in the Transport Table includes bits indicating whether it is active (a transaction is in process), the device is performing as the master, and whether it is Broadcast.

When the transport table entry is for the master end of the pipe, the retry count and retry timer are tracked. When the retry timer lapses, the request TPDU is resent and the retry counter

incremented. When the maximum retries is exceeded, the Transport Layer notifies its client that a fault has occurred. The counter shall be reset whenever a new transaction commences.

Most importantly, the (if acting as a master) last request payload or (if acting as a slave) the last response payload is cached along with the corresponding sequence number. This allows the transport master to resend the request if needed. Furthermore, if the device is the transport slave, the response can be resent if a repeated request is received. Repeated requests are identified by the repetition of the sequence number. The sequence number is initialized to a random number by the master device when the transport table is created and then incremented for each new transaction.

Table 24 – Transport Table Entries

Content	Description
Bits-1 Active	Set if the transport is ACTIVE (reset if acknowledge received)
Bits-1 Master	Set if the device is the MASTER (i.e., the side sending requests)
Bits-1 Broadcast	Set if the request packet is Broadcast
Unsigned-5 sequenceNumber	On Master, the sequence number for the outstanding request On Slave, the sequence number last acknowledged
Unsigned-8 TPDUHandle	On Master only, the handle from the TRANSMIT.request corresponding to lastTPDU
Byte[] lastTPDU	On Master, the last unacknowledged packet (or NULL if none) On Slave, the last acknowledged payload
Unsigned-8 retryCount	On Master only, Number of communication attempts
Time responseTimer	Timer to trigger a retry

6.4.3.4 Transport Layer Operating Sequence

The basic operation is shown in Figure 37 using the sequence performed when the Network Key is changed. The Write Network Key includes the key and the Absolute Slot Number indicating when use of the key shall commence.

The sequence begins with the Network Manager (the master in this example) generating an Acknowledged Broadcast Request NPDU and promulgating it to the network (see Figure 37, ①). Since it is a broadcast request, the transport table entry associated with the Broadcast Session entry in the Network Manager shall be used.

The Network Layer propagates the NPDU to all devices in the network. Each device, in turn, acknowledges the Write Network Key command (see Figure 37, ②). If necessary, the device creates the Transport Table Entry associated with the Network Manager-device Broadcast session.

The device replies and the acknowledge is enciphered using the unicast device-Network Manager session. This ensures intervening devices cannot, for example, spoof the acknowledge. However, to allow the Network Manager to correlate the acknowledge to the correct transport table entry, the broadcast bit remains set in the Transport Byte.

The Network Manager's Transport Layer receives acknowledges from all devices (retrying as needed). At the stipulated Absolute Slot Number, the Network Key in all devices roll over to the new value (see Figure 37, ③).

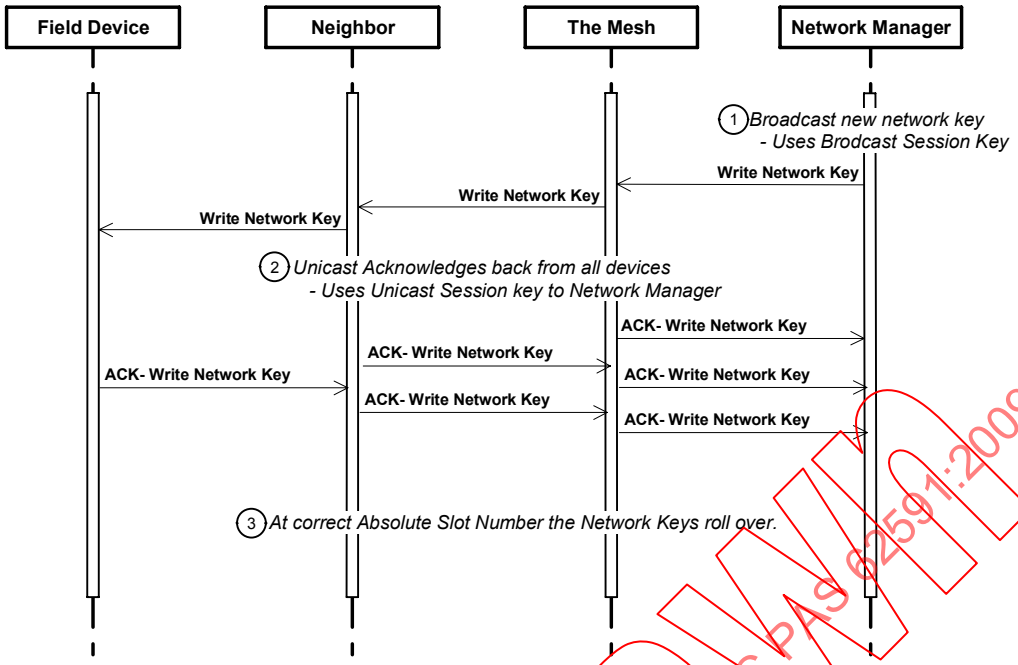


Figure 37 – Using Transport Layer to Change Network Key.

6.4.3.5 Transport Layer Operation

6.4.3.5.1 General

The Transport Layer builds upon the end to end secure sessions provided by the Security sublayer. The security sessions establish a connection between two devices. For each security session there can be two one-way transport pipes (i.e., the field device is the slave for one of the pipes and the master in the other). For each security session the corresponding transport table entries shall be created automatically or when the transport pipe is first used.

To summarize, the basic acknowledged operation consists of:

- The master generating a request;
- The master's Transport Layer tagging it with a sequence number;
- The request's propagation through the network;
- Reception by the destination (slave) device;
- Processing of the request and the generation of the response;
- The slave propagating the response back to the master using the same sequence number.

During this process the master will use the response timer to trigger retries as needed to ensure delivery of the request and reception of the slave's response. An error is signaled if the retry counter exceeds its maximum value.

6.4.3.5.2 Master Request Generation

6.4.3.5.2.1 General

When the client layer invokes TRANSMIT.request SP, the transportType parameter is inspected and the most significant three bits of the Transport Byte are set accordingly.

6.4.3.5.2.2 Un-Acknowledged Service

If the Acknowledged bit is reset, then the least significant five bits of the packetHandle are copied into the Sequence Number field. Construction of the NPDU is completed and it is

passed to the security sublayer for transmission. The TRANSMIT.confirm SP is invoked and the Network Layer transaction is complete.

6.4.3.5.2.3 Acknowledged Service

If the Acknowledged bit is set, then the corresponding Transport Table entry shall be located or, if it does not exist, the entry shall be created.

If there is already an unacknowledged packet pending, one already buffered up for this transport pipe, and no more buffers are available, then, the TRANSMIT.request fails and TRANSMIT.confirm SP is invoked to signal the error, and the request is discarded.

When the transport pipe is available, the sequenceNumber from the table entry is incremented and the least significant five bits are copied to the corresponding field in the transport Byte. The NPDU shall be then passed to the security sublayer for transmission. The packet shall also be buffered for possible future retries.

The Active flag is set in the Transport Table entry and the retryCount is initialized to 0. The replyTimer is initialized to maxReplyTime and started to await the response.

6.4.3.5.3 Master Retries

When a replyTimer expires, a retry shall be generated. The master shall increment the retryCount. If the count is exhausted (i.e., it exceeds maxRetries), then the TRANSMIT.confirm SP is invoked to notify the client layer of the failure. The Active flag is reset and the packet buffer released.

Otherwise, the saved copy of packet (i.e., the lastTPDU) is resent with the same Transport Byte as used in the original packet.

6.4.3.5.4 Propagation to the Destination Device

When received from the Transport Layer, the packet is processed by the Security Sublayer and propagated to the Data-Link Layer. From there, it is sent across the network and after possible several hops the packet arrives at its destination(s). During transit, the NPDU (including the Transport Byte) are enciphered to prevent their molestation.

6.4.3.5.5 Receiving the Request NPDU.

6.4.3.5.5.1 General

When the device receives a request NPDU from a correspondent master, it shall inspect the Transport Byte.

6.4.3.5.5.2 Un-Acknowledged Service

If the Acknowledged bit is reset then the TRANSMIT.indicate SP is invoked with the data field and the sequenceNumber set to the value in the Transport Byte. The Transport Layer transaction is complete.

6.4.3.5.5.3 Acknowledged Service

However, if the Acknowledged bit is set then the sequence number shall be validated. First, the correct Transport table entry shall be found by locating the session associated with the correspondent address. There are generally two sessions for each correspondent address (one Broadcast and the one Unicast). Using the Broadcast bit, the correct session is selected and the Transport table entry is accessed.

If the transport table entry does not exist, then it shall be created. The Active bit is set, the Master bit reset and the sequenceNumber is set to one less than that in the current packet.

If the transport table entry already exists, then the packet's sequenceNumber is compared to that found in the Transport Table. If they are the same, then the packet is a retry and the buffered response is re-sent. If the sequenceNumber is not one greater than that in the Transport table entry, then the packet is discarded.

If the transport table is new or the sequenceNumber is one greater than that in the Transport table entry, then the TRANSMIT.indicate SP is invoked with the data field and the sequenceNumber set to the value in the Transport Byte and the transportType set accordingly.

6.4.3.5.6 Generation of the Response NPDU

Upon reception of a TRANSMIT.indicate containing a request packet, the client layer shall process the request and provide a response. Once the response is prepared, the client layer shall invoke the TRANSMIT.response SP with the same sequenceNumber as in the request. Furthermore, the response shall set the Broadcast flag identically to that in the TRANSMIT.indicate.

Once the TRANSMIT.response SP is invoked, the correct Transport table entry shall be found by locating the security session associated with the correspondent address. There are generally two sessions for each correspondent address (one Broadcast and the one Unicast). Using the Broadcast bit the correct security session is selected and the Transport table entry is accessed.

The sequenceNumber provided in the TRANSMIT.response is compared to that found in the Transport Table. If the sequenceNumber is not one greater than that in the Transport table entry then the SP fails and the packet is discarded.

Otherwise, the packet is buffered and the sequenceNumber in the Transport Table updated. The device will use the buffered payload to repeat replies in the event of duplicate requests. Next, the NPDU shall be passed to the security sublayer for propagation to the transport pipe's master.

6.4.3.5.7 Master Collecting the Acknowledgement.

6.4.3.5.7.1 General

When the device receives a response NPDU, it shall inspect the Transport Byte.

6.4.3.5.7.2 Un-Acknowledged Service

If the Acknowledged bit is reset, then the TRANSMIT.confirm SP is invoked with the data field and the sequenceNumber set to the value in the Transport Byte. The Transport Layer transaction is complete.

6.4.3.5.7.3 Acknowledged Service

However, if the Acknowledged bit is set, then the sequence number shall be validated. First, the correct Transport table entry shall be found by locating the session associated with the correspondent address. There are generally two sessions for each correspondent address (one Broadcast and the one Unicast). Using the Broadcast bit, the correct session is selected and the Transport table entry is accessed.

If the transport table entry does not exist, then the packet is discarded.

If the transport table entry already exists, then the packet's sequenceNumber is compared to that found in the Transport Table. If they are the same then the packet completes the Transport Layer transaction. The TRANSMIT.confirm SP is invoked with the data field and the packetHandle set to the value from the TRANSMIT.request. TRANSMIT.confirm status is set appropriately.

Finally, the request packet buffer is released, the Active bit reset and the replyTimer disabled.

6.4.4 Wireless Network Layer Operation

6.4.4.1 Overview of Network Layer Behavior

6.4.4.1.1 General

A WirelessHART enabled device progresses through a series of states starting in the Idle state and continuing until the device is Operational and a full participant in the mesh network. There are six principal states and they are briefly described in Table 25. These states are discussed in more detail in the following subdivisions of 6.4.4.

Table 25 – Definitions of Network Layer States

State	Description
Idle	The device is quiescent and its wireless transceiver is not active. It has no knowledge of the WirelessHART network
Joining	The device is listening for the network, attempting to acquire an advertisement and requesting admission to the network
Quarantined	The device has successfully joined the network but only has a security clearance to talk with the Network Manager. It is not available or allowed to perform data acquisition or control functions or otherwise communicate with the Gateway
Operational	The device can be accessed by Host Applications via the Gateway. It is integrated in the system's operation
Suspended	The device is quiescent. All of its network tables are intact
Re-synching	The device is listening for the network. After identifying the slot time and ASN, it will begin issuing Keep-Alives to reconnect to its neighbors

The state transition diagram is shown in Figure 38 below. This diagram depicts the events that allow the device to start-up, locate and join the network, and finally progress from being a new network member to becoming a fully operational network device. In addition, the diagram shows the device (and network) being suspended and re-synchronized.

Furthermore, the device can be forced to disconnect (e.g., to be removed from the process for depot-level maintenance) or to re-join at any time.

6.4.4.1.2 Idle

After a device reset or on power up the device should enter the Idle state.

While in this state:

- the device shall not attempt to communicate wirelessly; and
- initial provisioning of the device should be perform using the maintenance port;
- the Join Key and Network ID are normally written to the device.

The device may stay in this state until instructed to initiate the Join process.

6.4.4.1.3 Joining

Once initiated, the Join process can have three outcomes: success, failure, or be aborted (e.g., by the reception of a Disconnect Command).

To be successful the device shall:

- locate the network;
- synchronize to the network;
- capture an Advertise packet;
- request Admission to the network from the Network Manager; and
- receive Network Keys and a Network Manager session; and
- be provisioned with superframe, graphs and links.

Once one or more Advertise packets have been received, the device shall request admission using the Join Key. The Join request may be answered by the Network Manager immediately or the device may need to retry the request.

A failure shall occur if the device sends an excessive number of join requests without receiving the security keys and session parameters from the Network Manager. In this case, the device shall return to the Idle state.

Once the keys and Network Manager session are established, the device shall pend on reception of a normal frame, graph and links. Once this occurs, the device has joined and is quarantined.

6.4.4.1.4 Embraced-Quarantined

Quarantined devices are limited network partners. Devices should remain quarantined until approved for deployment into the network application. Approval may take the form of the operator authorizing the use of the device in his system and placing the device into service. This quarantine step parallels normal practices in the process industry and adds an additional layer of security (if desired) to the join process

While quarantined, the device should only be enabled to source and sink packets (i.e., the device cannot forward packets). Normal network reports (e.g., neighbor reports) shall be generated.

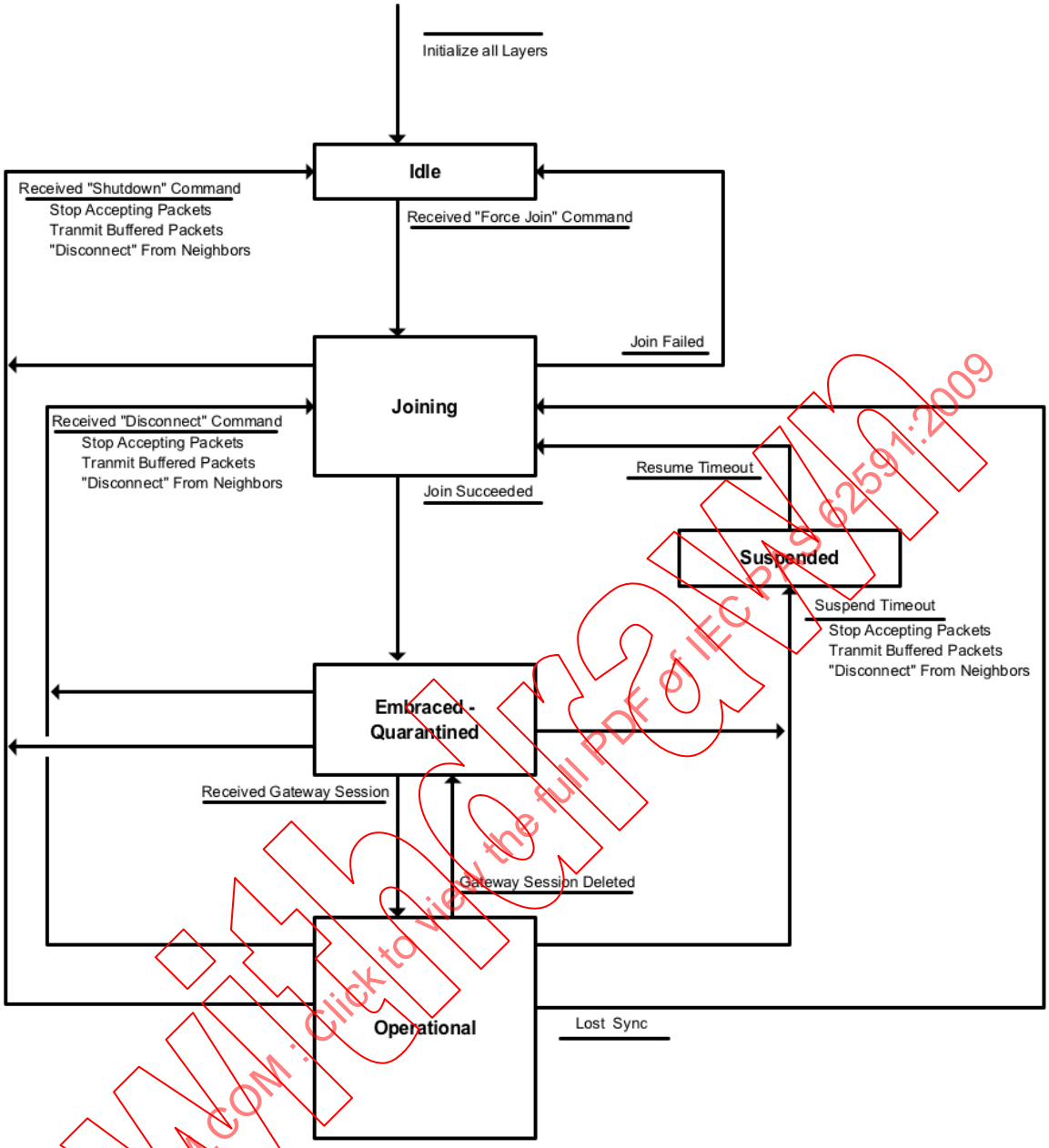


Figure 38 – WirelessHART Network Layer Operation

6.4.4.1.5 Operational

The device becomes operational upon the reception of the Gateway session parameters. This allows host application access to the device and begins its interaction with it.

Once the device enters the operational state it is a full network partner and shall begin performing the mission designated by its configuration. Based on its configuration parameters, the device shall request bandwidth, for example, to periodically publish data. Request/response traffic with host applications is enabled.

Since the device is a trusted full network partner, it may actively support operation and grooming of the network. This includes routing packets onward through the network to their final destination. The device also participates in advertising and supports new devices joining the network.

6.4.4.1.6 Suspended

When quarantined or operational, the device may be suspended (along with the network). This places the device into a quiescent state for the time specified. This is normally a brief amount of time and done, for example, for safety purposes during mining blast operations.

When in this state, the device continues to track elapsed time while leaving the radio disabled. Once the specified time has elapsed, the device progresses to the Re-syncing state.

6.4.4.2 Network Layer Data Model

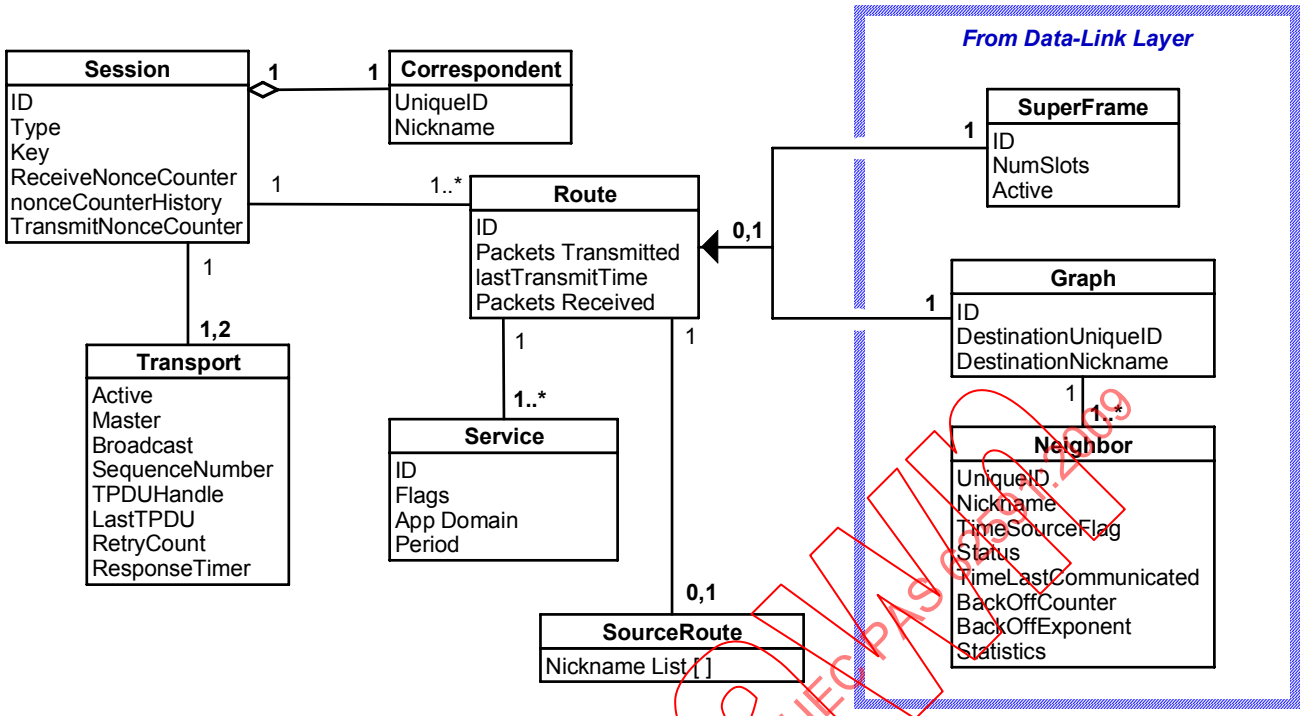
6.4.4.2.1 General

All devices maintain a series of tables that control the communications performed by the device, supply routing information, support end to end acknowledgements and ensure the privacy of the communications. The communication tables and the relationships between them are shown in Figure 39 below. Within the device, the Session table is central and contains references to all correspondents the device can successfully communicate with. The tables controlling communication activities include the following:

- Session table. All communications center around the security session. The session table entry establishes a secure pipe between the device and a specific correspondent device. No communication to a correspondent device is possible without a security session and all devices shall support multiple security sessions.
- Transport table. The transport table is used to support end to end acknowledge transaction with automatic retries. There are generally two per session.
- Route table. The route table serves as the locus for adding routing information to a new NPDU being generated as the result of a TRANSMIT request.
- Source Route table. A source route list is attached to some routes. The source route (when present) contains up to 8 device addresses tracing the route from the device to the correspondent.
- Service table. This table indicates the Route associated with a service allocated by the Network Manager. The same Route may be used by more than one Service.

NOTE Graphs and Neighbors are also discussed in the TDMA Data-Link Layer.

- The Graph table. Graphs are used to route messages from their source to their destination.
- The Neighbor table. The neighbor table is a list of all devices that the device may be able to communicate with. The device does not know the entire route rather there are references from the graph to the neighbor that indicate the legal next hop Data-Link destinations. Neighbors with links to the device are listed first, followed by detected (discovered) neighbors.



NOTE Although specific implementation of data and configuration storage is left up to the designer, the descriptions of the fields are critical understanding device requirements. Some fields described in the tables in 6.4.4.2 may be calculated or derived from other information, and do not necessarily occupy space on the device.

Figure 39 – Wireless Network Table Relationships

Devices shall support the minimum number of tables entries shown in Table 26.

Table 26 – Minimum Session Table Space Requirement

Description	Minimum Required
Sessions	8
Correspondent Device	1 per Session
Transport	2 per Session
Routes	8
Source-Routes	2
Services	16

6.4.4.2.2 Route Table

The route table is associated with a security session and, thus, a correspondent address. This is used to aid in the selection of the graph used to communicate to the correspondent. Statistics are also gathered to allow the Network Manager to optimize communication resources over time.

Table 27 – Route Table Entries

Content	Description
Unsigned-8 routeID	Route ID
Bit-8 routeType	Bitmap: {Maintenance (default), Publish, Block Transfer, Event}
Unsigned-16 NumPktsTransitted	Number of Packets Transmitted
Time lastTransmitTime	Time last NPDU generated for this Route
Unsigned-16 NumPktsReceived	Number of Packets Received on this Route. This is updated based on the Graph ID in the NPDU

6.4.4.2.3 Service Table

The primary purpose of the service table (see Table 28) is to track the services allocated to the device. When the device deletes a service, the Network Manager de-allocates the network resources supporting the service.

Table 28 – Service Table Entries

Content	Description
Unsigned-8 serviceID	Service ID
Bits-8 serviceFlags	Flags (e.g., source, sink, intermittent)
Enum-8 serviceDomain	The Application domain using this service
Time servicePeriod	For cyclical communications (e.g., Publish Data), packets are generated with this period. Latency for cyclical communication shall not exceed 1/3 the period For intermittent communications specification, this is the required latency
Unsigned-8 routeID	The routeID is returned by the Network Manager when it allocates the serviceID. More than one service can share the same routeID

6.4.4.3 NPDU Management

6.4.4.3.1 General

One of the core responsibilities of the Network Layer is the processing of NPDUs as they are received. There are three Network Layer clients that can source or sink an NPDU (see Figure 40):

1. The Application Layer is a packet sink (signaled using the TRANSMIT.indicate SP) and a source (invoking the TRANSMIT.request SP)
2. The Data-Link Layer that sources packets for consumption or routing by the Network Layer.
3. A Joining Device serviced via the Data-Link. When Proxy routing, the device acts as the NPDU destination on behalf of the Joining Device.

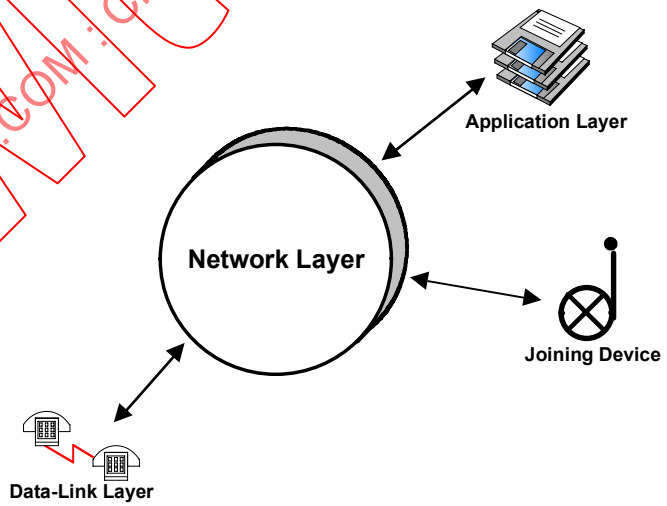


Figure 40 – NPDU Clients

6.4.4.3.2 Processing Packets from the Application Layer

When a transmit request is received from the Application Layer, NPDU shall be constructed. The field shall be set according the procedures indicated in Table 29.

Table 29 – NPDU Construction

Field	Procedure
Destination	Set to value in Application Layer request (long, short, broadcast)
Source	Set to long address if Nickname uninitialized
ASN Snippet	Set to current ASN value
TTL	Normally set to default value
Graph ID	Set based on results of route selection
Control	Set the address bits based on source and destination address sizes. Destination addresses are indicated in the TRANSMIT.request. Source address shall be the device's EUI-64 address until the device's Nickname is initialized Set the routing bits based on results of route selection
Proxy Route	Set based on results of route selection
Source Route(s)	Set based on results of route selection

The route shall be chosen as indicated in Table 30. Once the route has been identified the Graph ID is copied into the NPDU.

Except when testing routes, packets are normally graph routed. However, if a Source-Route is attached, then the source route addresses shall be included in the NPDU. Unused source route address entries in the NPDU shall be set to 0xFFFF. If source routing is used the Graph ID should also be valid.

Table 30 – Default Route Based on Priority and Transport Type

TRANSMIT.request Field		Resulting Application Domain	Default Route Type
Priority	Transport		
Command	Don't Care	Maintenance	First "Maintenance" route to the Network Manager
Data	Don't Care	Publish	First "Publish" Route to the Gateway
Normal	Transfer	Block Transfer	First "Block Transfer" Route as indicated by destAddress
	! Transfer	Maintenance	First "Maintenance" Route as indicated by destAddress
Event	Don't Care	Event	First "Event" Route to the Gateway or Network Manager as indicated by destAddress

If a route matching the Application Domain is not available, then the first route to the destAddress shall be used.

6.4.4.3.3 Processing Packets Received from the Data-Link

6.4.4.3.3.1 General

The Network Layer will be passed NPDUs as they are received by the Data-Link Layer. The Network Layer shall route these packets to the correct destination: back to the Data-Link Layer, onward to the Joining device, or up to the Application Layer.

6.4.4.3.3.2 Packets Addressed to the Device

If the device is the NPDU's final destination, then the NPDU is authenticated and deciphered, discarding if necessary. The Transport Layer shall be invoked which, in turn, forwards the payload to the Application Layer.

6.4.4.3.3.3 Packets with a Broadcast address

When the NPDU is received and the destination address is the broadcast address, then the NPDU is authenticated and deciphered, discarding if necessary. The Transport Layer shall be invoked which, in turn, forwards the payload to the Application Layer. In addition, the NPDU shall be forwarded as described in 6.4.4.3.3.4.

Some broadcast commands (e.g., Command 961) may result in responses from a large number of devices. For these commands, a random back-off time is chosen to delay the device response and minimize the flood of simultaneous responses. The back-off time is a random value chosen between zero and BcastReplyTime.

For broadcast commands that should generate a single, unique response (e.g., Command 21), the response shall be generated immediately.

6.4.4.3.3.4 Forwarding Received Packets

Prior to forwarding an NPDU, the device shall check and update the TTL. If the TTL count is exhausted, the NPDU shall be discarded. Next, the ASN Snippet is inspected and, if the maxPacketAge is exceeded, the packet shall be discarded. While inspecting the ASN Snippet, the timeout value to be passed to the Data-Link shall be calculated.

Upon confirming the packet does not need to be retired, its destination address shall be inspected.

- If the NPDU contains a proxy address that matches the device's, then final destination is a joining device and it is the device's responsibility to deliver the NPDU. The device shall route the packet to the joining device by passing the NPDU back to the Data-Link indicating the Data-Link destination address is the joining device's. The TTL and ASN Snippet are inspected as indicated in the "Forwarding Received Packets" see next bullet items.
- If the Unicast NPDU destination address matches a neighbor, then the NPDU shall be routed directly to the neighbor. In this case, the packet is passed to the Data-Link for propagation directly to the final destination. If this fails (an error is returned from the Data-Link), then the packet shall be Graph routed (if possible).
- Otherwise, the packet shall routed onward based on either the Graph ID or the Source-Route contained in the NPDU. The routing actions to be performed are depicted in Table 31 below. When analyzing a received NPDU, the fields in the table shall be evaluated from left to right (i.e., starting with whether the NPDU contains source route information). A "No" in the Graph column means that the Graph ID is invalid (i.e., the Graph field does not match either a Graph ID or a Superframe ID found in the field device). The combinations NOT depicted are illegal and result in the NPDU being discarded.

The "Action" column indicates the routing action to be performed based on the state indicated in the first four columns. This results in either a specific address being provided to the Data-Link for the next hop or for the Data-Link to broadcast the NPDU. Two errors are possible: the NPDU reaches the end of a source route without reaching its final destination; and the NPDU reaches the end of a graph route without reaching its final destination. These errors can only happen when the final destination is Unicast and the packet was not being broadcast across the Data-Link.

Table 31 – Routing of Forwarded Packets

Routing		Destination Address		
Source	**Graph**	**DLPDU**	**NPDU**	**Action**
Yes	Yes	Unicast	Don't care	Forward the NPDU to (1) the next address in the source route; or (2) broadcast to the Superframe (Graph ID contains the Superframe ID) if source route exhausted
		Broadcast	Don't care	Continue broadcasting to the graph. Use broadcast link in the Superframe (Graph ID contains the Superframe ID)
	No	Don't care	Unicast	Forward the NPDU to (1) the next address in the source route; or (2) signal a source route error
			Broadcast	Forward the NPDU to (1) the next address in the source route
No	Yes	Broadcast	Don't care	Graph ID is Superframe ID. Continue broadcasting the NPDU using broadcast link in the superframe
		Unicast	Unicast	Forward NPDU along the Graph (using any normal link to neighbor on the Graph). If end of Graph signal graph route error
			Broadcast	Forward NPDU to all neighbors in the graph. NPDU discarded at end of Graph

NOTE When DLPDU is broadcast, then the Graph ID holds the Superframe ID used to identify the corresponding broadcast links.

6.4.5 WirelessHART Procedures

6.4.5.1 Initializing a WirelessHART Network

Prior to forming the network, the Network Manager shall be provisioned with the Network ID. Using this ID and a supply of security keys, network formation can be initiated. This begins with the Network Manager creating a secure and private connection with the Gateway. As part of its initialization sequence, the Network Manager will download to each of the Gateway's Access Points:

- The network management superframe supporting base bandwidth required to monitor and service the network;
- The network graph supporting upstream traffic to the Network Manager;
- The Join superframe and Join Links allowing new devices to join the network; and
- Dedicated and shared Links (both transmit and receive) supporting management of devices, the transport of health reports, and communication of alarms (e.g., path down).

In general, the Gateway's Access Points are configured to be active in every slot. This maximizes the advertising, network management, and join packets available to the network. Once the Network Manager enables the first superframe, ASN 0 is established (i.e. the network is born). Once the Gateway's Access Points begin transmitting Advertise packets, devices can join the network and the network begins forming.

There are three components of network formation: *advertising*, *joining*, and *parameter negotiation*. As part of advertising, Network Devices that are already part of the network may send packets announcing the presence of the network. Advertise packets include the Network ID, ASN, join frames and join links. Devices that are trying to join the network listen for these packets. Once an Advertise packet for their network is heard, the new device can attempt to join the network. The join sequence is described in 6.4.5.2.

As the device joins the network, both the device and the Gateway request bandwidth from the Network Manager. For example, the device asks for bandwidth to publish process data and the Gateway requests bandwidth to support request/response traffic. The Network Manager uses these service requests to gauge and manage the available bandwidth. Assuming there is sufficient bandwidth, the newly joined device is allocated superframes and links according to

the requests. If bandwidth becomes constrained, then the Network Manager may reduce the bandwidth requested or refuse to allocate any at all.

6.4.5.2 Joining

6.4.5.2.1 Overview

"Joining" refers to the process used by the device to obtain access to the network and to become integrated into it. The key steps in the joining process include the following:

- Periodic Advertise packets by existing network members to allow the network to be identified. The Advertise packet also includes sufficient information for the new device to synchronize to the network and communicate on the correct link (i.e., on the right slot number and channel).
- Monitoring by the new device to locate and synchronize to the network.
- Establishing a secure channel between the new device and the Network Manager. This is done using the Join Key to encipher the initial communications between the two devices. The Join Key can be different for every device.
- Verifying the trustworthiness of the new device. This is done several ways: The device's Identity (Command 0) and Long Tag are presented to the Network Manager. Furthermore, the Join Key is, in effect, a password. All of three of these items can be used when considering to allow the device into the network.
- Once the device is deemed trustworthy, it can be provisioned and allowed into the network. Initially the device can be quarantined until the plant operations are ready to begin utilizing the device.

In 6.4.5.2 an overview of the Join process is provided followed by detailed Network and Data-Link Layer join requirements.

6.4.5.2.2 Overview of the Join Sequence

6.4.5.2.2.1 General

An overview of the Join Sequence is shown in Figure 41 below. For a new device to become operational, it shall be provisioned with Network ID; locate and synchronize with the network; petition the Network Manager for access; obtain session keys; and gain the bandwidth necessary to meet the obligations the device's configuration has imposed. The general progression that shall be followed for the joining device to become operational includes the following:

- Initial device provisioning consists of obtaining the Network ID and Join Key.
- The device shall begin listening for network traffic to allow it to synchronize to the network clock and identify potential parents.
- Next, the device presents its credentials to the Network Manager to demonstrate the device is trustworthy. The credentials include the device's identity and Join Key and, if these credentials are valid, the device is admitted to the network.
- Once the Network Manager has scrutinized the device's credentials and deems the device trustworthy, the Network Manager provides the first keys (Network Manager Session Key and the Network Key) to the joining device.
- Once security requirements for new devices have been met the Network Manager proceeds to integrate the device into the network. This is accomplished by provisioning the device with normal superframes and links.
- The Network Manager may choose to leave the device Quarantined. In this case the device can participate in the network but does not have a Gateway session.

- Once the quarantined device obtains a session with the Gateway it becomes operational. It then begins acquiring the bandwidth and communication resources required to publish process data and events as dictated by its configuration.

Each of these steps are discussed in detail in the following paragraphs.

6.4.5.2.2.2 Initial Device Provisioning

Prior to attempting to join the network the device requires two pieces of data: the Join Key and the Network ID. The Network ID identifies the network the device is to Join and the Join Key is the network password that will allow the device to join the network. These two items should be written to the device via the device's maintenance port.

Once initial provisioning is complete, the device can be immediately configured to attempt joining the network. Alternatively, the device can be mounted in the process and then the end user can use a device maintenance tool to place the device into join mode.

The maintenance tool may also be used to monitor the join process allowing the operator to intervene (if necessary).

6.4.5.2.2.3 Listening for Network Traffic

Once the join has been initiated, the device's Data Link is placed into search mode to synchronize to the network and receive Advertise packet. While in this mode, the device will identify neighbors (advertising or not) and gather neighbor statistics (e.g., average signal level). Once one or more advertising neighbors have been identified, the joining device selects one of them to join through. The first join attempt shall be made via the neighbor demonstrating adequate Receive Signal Level and indicating the lowest Join Priority.

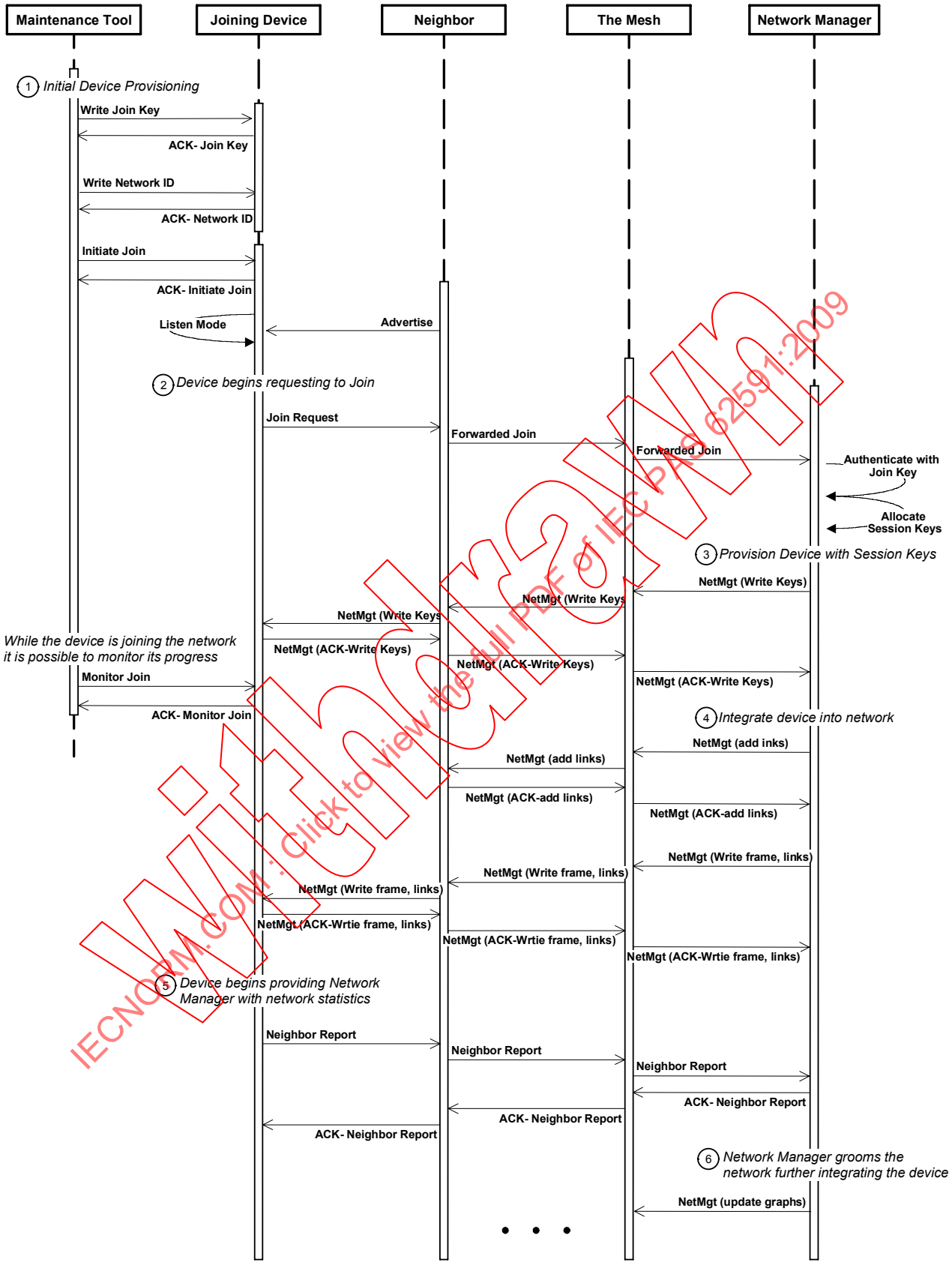


Figure 41 – Join Sequence

6.4.5.2.2.4 Presenting Credentials

Once synchronized to the network, the device generates a join request (see Figure 41,②) and sends it to the Network Manager. The join request and its encipherment contain the device's

credentials for the Network Manager to inspect prior to allowing the device to join the network. The device's objective is to obtain a Session Key (for further communication with the Network Manager) and a Network Key (for Data-Link Layer device to device, one hop security and authentication). There are three key credentials to be presented plus the list of neighbors detected by the device:

1. the Device's Identity (i.e. Command 0 response);
2. its Long Tag (Command 20 response); and
3. the list of Neighbors detected by the device (Command 787 response).
4. the device's Join Key (i.e., the device's password to the network).

The first two are contained in the join request's payload. The Join Key is used as the session key for all communications until the device receives Session and Network Keys from the Network Manager. Consequently, successful authentication by the Network Manager confirms the device has the correct Join Key.

Since the joining device does not know the Network Key, (i.e., the Data-Link key) the well-known key is used for communicating the DLPDUs with the advertising neighbor (i.e., the device's prospective parent). Once the prospective parent receives the packet, it is forwarded through the mesh using the Network Key the same as any other DLPDU.

Once the join request is sent, the device starts a response timer in the same fashion as used in the Transport Layer. When this timer lapses, another join request is generated to the next available advertising neighbor. Join requests are sent until maxJoinRetries is exceeded.

6.4.5.2.2.5 Getting the First Keys

When the Network Manager receives a join request, it shall confirm the request was from a trusted device. A trusted device satisfies the following criteria:

- Has a trusted identity (i.e., the right device name);
- Is using an appropriate Join Key (the right password); and
- Properly combines the device name and password with each other.

The join request has sufficient information to satisfy all of these criteria. Identity is established using the binary information in the Command 0 or using the Long Tag in Command 20.

NOTE There is a wide range of methodologies and security techniques to establish whether a joining device is to be trusted. Selection of the strategy employed by Network Managers to confirm the join device is a trusted devices is beyond the scope of this PAS.

If the device's credentials are in order, then the join request authenticates, the Network Manager allocates Session Keys, the device's Nickname (i.e., the 2-byte short address) and writes these along with the Network Key back to the joining device. Like the join request, the NPDU is enciphered using the Join Key. The NPDU containing the keys should be routed to the joining device via its prospective parent.

The device acknowledges the write commands received from the Network Manager. The acknowledgement packet uses the new Session and Network Keys.

6.4.5.2.2.6 Device Integration into the Network

Now, the device has keys and a network ID and is, albeit awkwardly, able to communicate with the Network Manager. At this point in the Join sequence, its communication requires the Network Manager to use proxy routing to reach the joining device. The next step is for the Network Manager to integrate the joining device more tightly into the network. This includes the following:

- Providing the device with at least two time-source parents;

- Updating the communication tables in the device's parents;
- Transferring the device's communication from join links to normal links.

Examples of the associated communication activity are shown in Figure 41, starting at ④. This communication activity will continue for some time and result in the Network Manager writing a series of frames and links to the joining device. After this, the device is integrated into the network and no longer may use join links for communication.

The device has now "Joined" the network.

NOTE The Network Manager shall not delete or suspend any join links while a device is in the process of joining.

6.4.5.2.2.7 Quarantine

When a device is a member of the network and has no session with the Gateway, the device is "Quarantined". While quarantined, the device can only communicate with the Network Manager and shall not publish process data. The device shall operate normally, conveying DLPDUs received from neighbors, generating neighbor reports and network statistics. The Network Manager can modify the network communication configuration as needed to groom the network.

As soon as the device enters the Quarantine state, it begins generating health reports (see Figure 41,⑤, above). Upon entering this state, the HealthReport timer is initialized to HealthReportTime (see Table 18 above) and the timer is enabled.

The device generates health reports whenever the HealthReport timer lapses. Health reports are transmitted at the "Process-Data" priority level. Following every health report the HealthReport timer is reset to HealthReportTime.

After joining the network, the first health report shall be generated after identifying (at least) three neighbors or when the HealthReport timer expires, whichever comes first. If multiple Advertise packets were captured the first report shall be generated immediately after entering the Quarantine state.

Health reports consist of the response PDUs for Command 779, 780, and 787. Using Command 780, statistics on all linked neighbors are returned. Command 787 reports on all detected neighbors that do not have links to the device. Depending on the network and number of devices, multiple packets are normally generated. Health reports are aggregated into as few NPDUs as possible.

Depending on the Network Manager's security strategy, it may immediately write the Gateway session to the quarantined device or leave the device quarantined for a time. In any case, the device will remain quarantined until it receives a session allowing communication to the Gateway.

6.4.5.2.2.8 Becoming Operational

When a new Gateway session is written to the device, it becomes operational. Once this session is operational, a Gateway will normally begin filling the Gateway's data cache for device. This will result in a surge in request/response traffic to the device and may be several hundred transactions. This communication uses the Maintenance service. The maintenance service is owned and controlled by the Gateway. Upon receiving a session to a new device, the Gateway should request significant bandwidth (e.g., one request/response per second) to the device while it fills its cache then lower it to a normal rate (e.g., one request/response per 10 s).

In addition, external host applications (e.g., instrument management packages, process automation controllers, etc.) may begin communicating with the device. The Gateway shall increase and decrease the Maintenance service bandwidth guaranteeing responsive

communications with the field device. This communications rate should be similar to that found in FSK-based communications.

Once the device receives its Gateway session, it shall obtain enough bandwidth and communication resources to meet its responsibilities to the process automation system as a whole. Once operational, the device begins requesting bandwidth for the process data (the Publish service) and alarms (the Event service) it shall publish.

Once operational the Network Manager will continue adding or refining superframes and links used by the (now operational) device. Although the Network Manager will continuously adjust communication resources on its own in response to changing communication requirements, normally the operational device will trigger Network Manager activity itself by requesting resources from the Network Manager. This is covered in a separate procedure.

6.4.5.2.3 The Network Layer Join Process

6.4.5.2.3.1 General

The join process is managed from the Network Layer using two cascaded state machines. The Network Layer state machine enforces the high-level join procedure (including retries for security purposes) while the Data-Link state machine is focused on synchronizing the device to the communication slot times. The primary objective of the Join Process (at the Network Layer) is to receive admission to the network and obtain a frame (with links) allowing the device to communicate reliably with the Network Manager.

Upon beginning the process, the Data-Link is signaled to enter its join process and begin actively searching for the network.

6.4.5.2.3.2 Searching

While in the searching mode, the Network Layer waits for reception of an Advertise packet. While in this mode, the Data-Link is searching for the network, synchronizing to it and receiving DLPDUs.

Once the ADVERTISE.indicate signal is received from the Data-Link, the Network Layer sequences to the "Got an Advertising Neighbor" state.

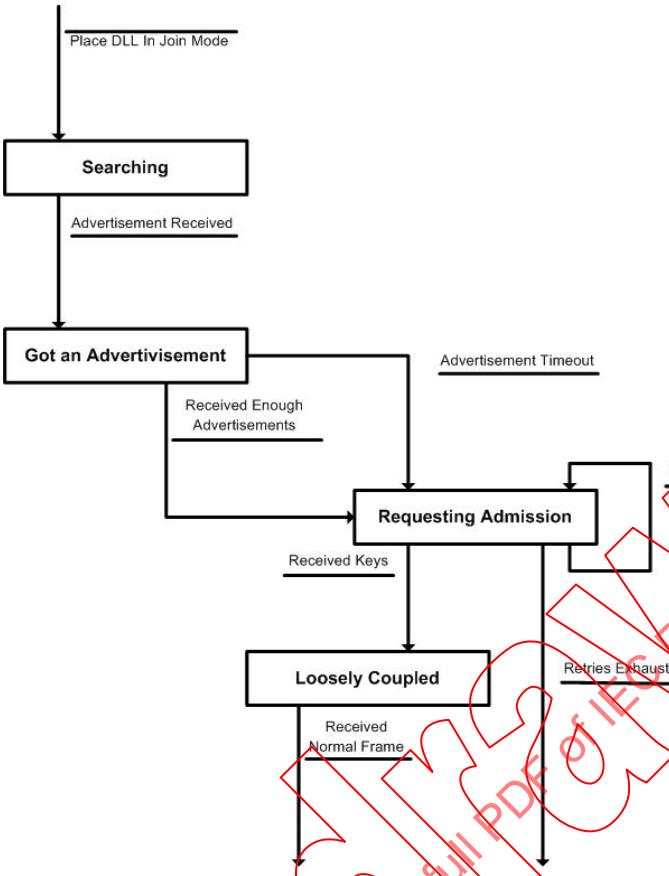


Figure 42 – Network Layer Join Procedure

6.4.5.2.3.3 Got an Advertising Neighbor

Upon entering the "Got an Advertising Neighbor", the AdWaitTimer is initialized to AdWaitTimeout (see Table 18 above) and started. The device continues to wait for additional Advertise packets to be received. When the desired number of different Advertise packets has been received or the AdWaitTimer times out, the device moves to the "Requesting Admission" state.

6.4.5.2.3.4 Requesting Admission

Upon entry into the "Requesting Admission" state, the device shall send a join request to the Network Manager, initializes the JoinRspTimer to JoinRspTimeout (see Table 18 above), and starts the JoinRspTimer. If the Network Manager responds by writing the Network key, the Network Manager session, and the device's nickname, the device progresses to the "Loosely Coupled" state.

Otherwise, upon JoinRsp timeout, the join request is issued again and the JoinRetry counter is decremented (see Table 18 above). Join Requests continue to be transmitted until the retries are exhausted or a response from the Network Manager admits the device into the network.

If the retries become exhausted, the state machine is terminated and exits with an error.

6.4.5.2.3.5 Loosely Coupled

Once the device has a Nickname and keys, it is able to communicate with the Network Manager. However, the device's connection to the network is tenuous at best (it only can talk via the shared join links). The device shall stay in the "Loosely Coupled" state until it receives a normal frame and links. If the device gets a frame, then it exits the Network Layer join

process successfully. Reliable communications between the Network Manager and the device is now ensured.

6.4.5.2.4 Data-Link Join Process

6.4.5.2.4.1 General

The Data-Link is triggered to enter the network search mode by the Network Layer. The objective of this mode is to synchronize the device's slot timing to that of the network. The first step in synchronizing is to begin actively searching for the network.

Figure 43 shows the state diagram of the DL network search procedure.

NOTE The join process is a device-level process and requires close coordination between the Network and Data-Link Layers.

6.4.5.2.4.2 Active Search

While in the "Active Search" state the device shall leave its transceiver on in receive mode while continuously listening for packets. When the Data-Link enters this state, it sets a timer to ActiveSearchShedTime to bound the network active search time. If this time lapses without identifying the network, the device transitions to the "Passive Search" state.

The device's transceiver remains on a channel for the interval indicated by the ChannelSearchTime value. When that time period lapses, the device shall switch to the next channel and resume listening. Channels are scanned sequentially until ActiveSearchShedTime lapses or a packet is received.

When a packet is received, the device sequences to the "Packet Received" state.

6.4.5.2.4.3 Passive Search

The Passive Search state is a reduced cycle, power saving mode of operation entered when the network cannot be located. In this mode, the device continues to prospect for network packet, just at a slower rate. The Device wakes up and listens for the interval indicated by the PassiveWakeTime. If no packet is received, the device turns off its transceiver and returns to low-power mode. This cycle repeats as specified by the PassiveCycleTime.

If a packet is received, the device transitions to the "Packet Received". The device can also be forced back into the "Active Search" state by the reception of the "Force Join Mode" command.

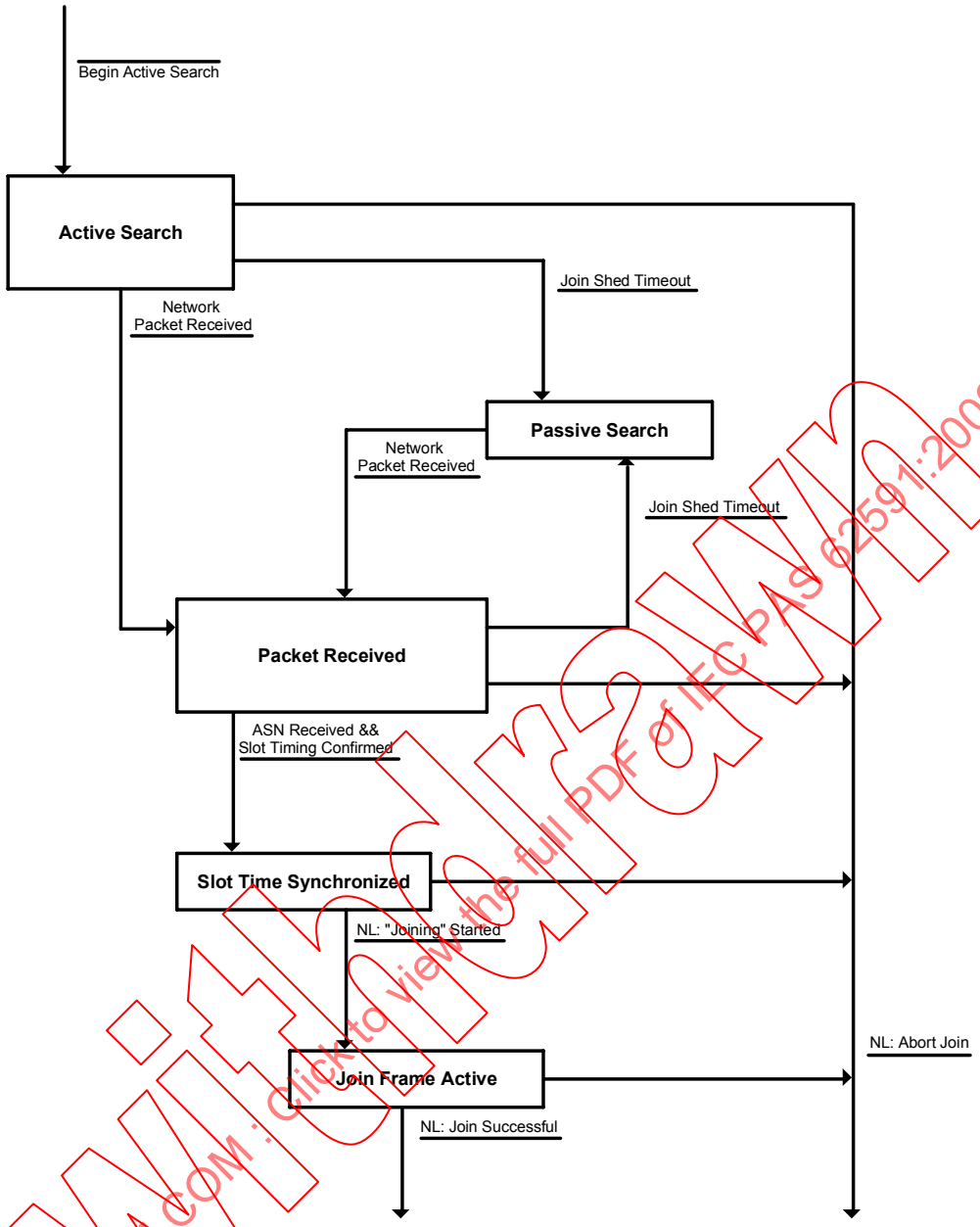


Figure 43 – Data-Link Layer Network Search Procedure

6.4.5.2.4.4 Packet Received

The objective of this state is to synchronize the device's slot timer to the network. This state is entered when a network packet is received (i.e., a packet with the correct Network ID).

NOTE Upon entering this state the ActiveSearchShedTime is restarted. If this time lapses without identifying the network, the device transitions to the "Passive Search" state.

If the DLPDU is not an ACK, then the start time of that packet is recorded and the device's slot time is established. Subsequently, the start time of all additional non-ACK packets are compared to the devices slot timing and statistics are compiled measuring the device's synchronization to the network slot time. The device is considered synchronized when the slot timing statistics converge.

While in this state, the device continues to capture network packets. As it does so, it shall update its neighbor table as normal. In addition, channels continue to be changed as

indicated by the ChannelSearchTime value. If an Advertise packet is received, the channels searched shall be limited to those indicated in the Advertise packet's Channel Map.

This device transitions to the "Slot Time Synchronized" state when (1) the slot time is synchronized, (2) the Absolute Slot Time has been aligned to the network. When these two criteria are met the Data-Link transitions to the "Slot Time Synchronized" state.

6.4.5.2.4.5 Slot Time Synchronized

Once this state is entered, the Data-Link can reduce its listening to slot times. As packets are received, the device's timers are updated keeping it in sync with the network.

At this point, the Data-Link is pending on the Network Layer to generate and propagate a join request. However, before the Network Layer will generate a Join request an Advertise packet shall be received.

When an Advertise packet is received, the Network Layer is signaled using the ADVERTISE.indicate SP. The device shall configure the Join frame(s) and links as indicated in the Advertise packets it receives. These frames remain disabled until join requests are generated. In addition, the Graph indicated in the Advertise packet is initialized and the connection to the advertising device created.

When a join request is received from the Network Layer, the device enables the join frame(s) and transitions to the "Join Frame Active" state.

6.4.5.2.4.6 Join Frame Active

The first join attempt shall be made via the neighbor demonstrating adequate Receive Signal Level and indicating the lowest Join Priority. Once the join request is propagated, the device shall begin issuing Keep-Alive packets, as needed, to maintain synchronization with Neighbors connected via Join Links. The device stays in this state until the Network Layer signals that the Join was successful or the join is aborted.

Before the join request is generated, the join frames received via the Advertise packets are enabled. Joined frames contain shared slots. Consequently, collisions with other joining devices are probable. To minimize collisions resulting from many devices trying to join simultaneously, the BOExp shall be initialized to four prior to transmitting the first join request.

6.4.5.3 Device leaving the network

In the network operation there will be times when a device is either suspended or disconnected from the network. When this occurs, the device shall go through a complete re-join sequence. Prior to departing the network, the device should not accept any additional packets and shall send a disconnect PDU to all of its neighbors. This is shown in Figure 44.

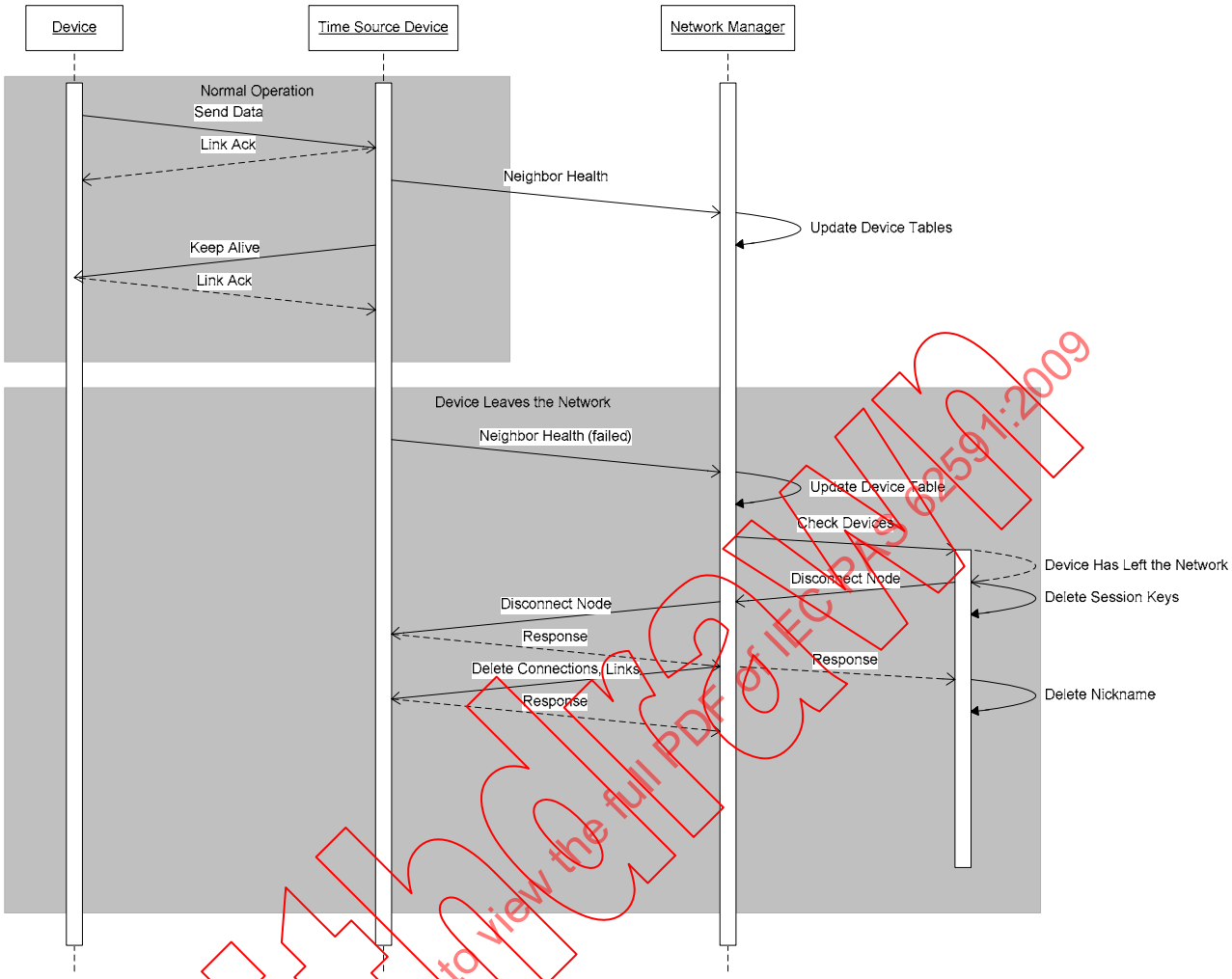


Figure 44 – Device leaving the network

6.4.5.4 Neighbor Discovery

The neighbor discovery process is used to learn of potential connectivity to new devices. The device maintains a list of discovered devices in its Neighbor table following the neighbors with links to the device. Network Devices periodically report neighbor information in their Health Reports using Commands 780 and 787. The Network Manager uses the information in the Health report to adjust the overall network graph and in some cases, adjust the schedule. This overall sequence is described in Figure 45.

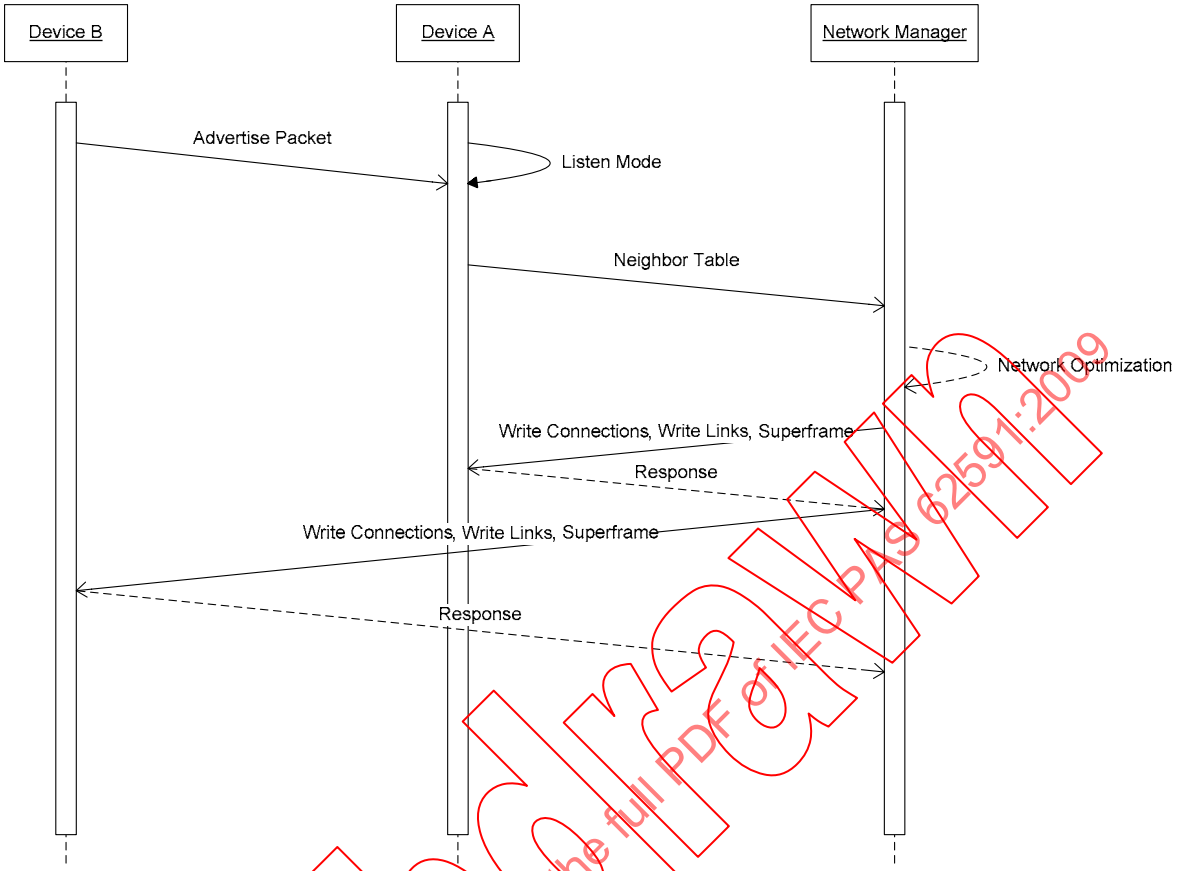


Figure 45 – Neighbor Discovery

6.4.5.5 Path Failure

Path failures are reported to the Network Manager when devices lose connectivity to neighbors. Figure 46 depicts three devices and shows that Device A and Device B have been successfully communicating (e.g., A published via B). However, interference or some blockage disrupts communication between A and B. Of course, the mesh allows A to continue responding via an alternate route (e.g., via Device C). Since communication was lost, A and B begin transmitting Keep-Alive packets to probe the connection. Communications continues to be problematic and, after the pathFailInterval lapses, a Path-Down Alarm (Command 788) is generated by both of them.

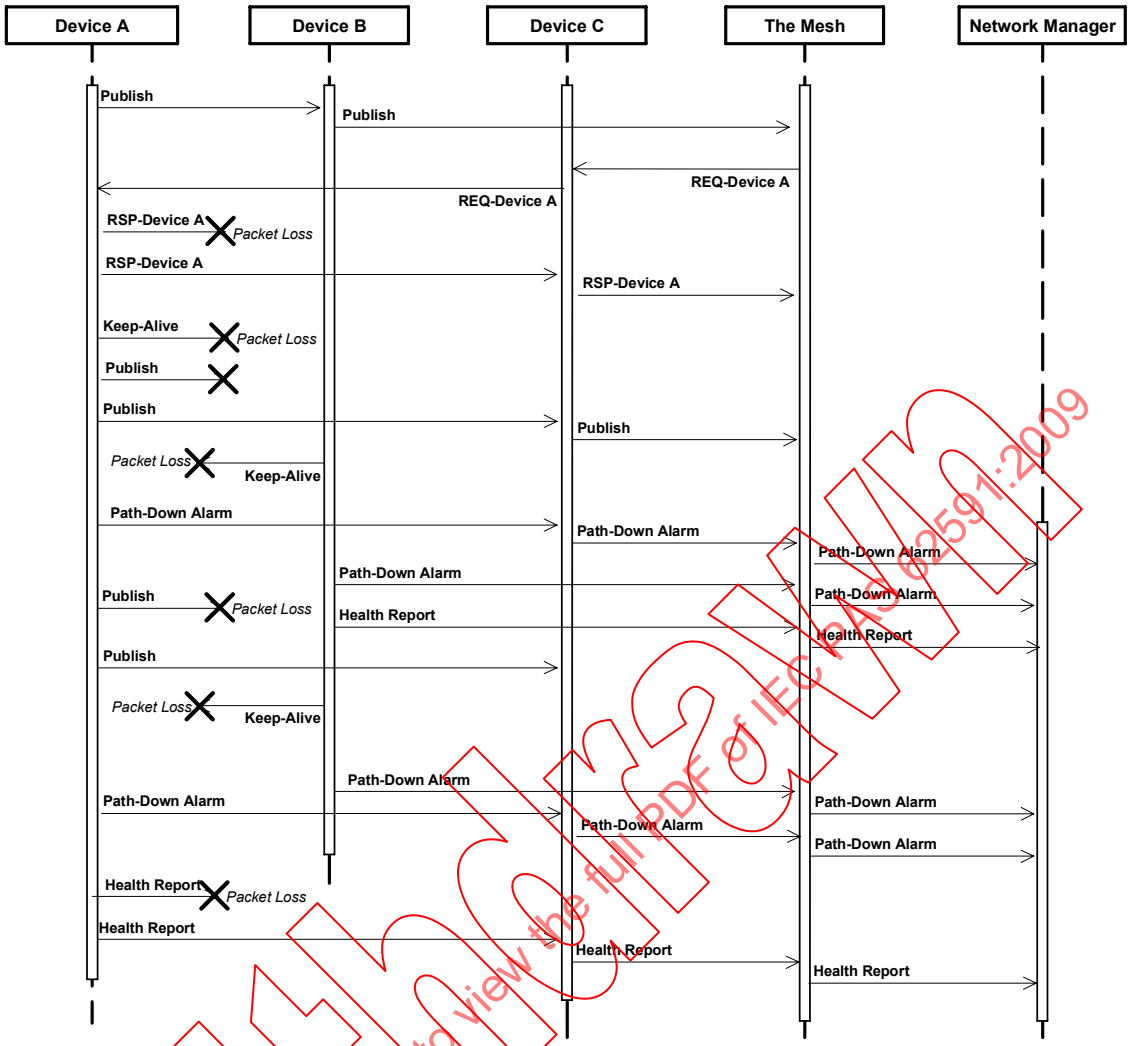


Figure 46 – Path Failure

As each device's HealthReport timer lapses, the devices generate health reports, which include indications of any problems the device is having with a neighbor. Notice that, since the devices joined the network at different times, the health reports do not occur simultaneously.

The devices continue trying to reestablish communications until the links between them are deleted by the Network Manager. It is common for broken paths to be restored as a temporary environmental effect passes. If the disruption persists, additional Path-Down Alarms will be generated when the pathFailInterval lapses again.

6.4.5.6 Changing the Network Keys

Changing the Network Keys is an important operation that shall be done periodically to ensure the overall integrity of the system. This is done as shown in Figure 47.

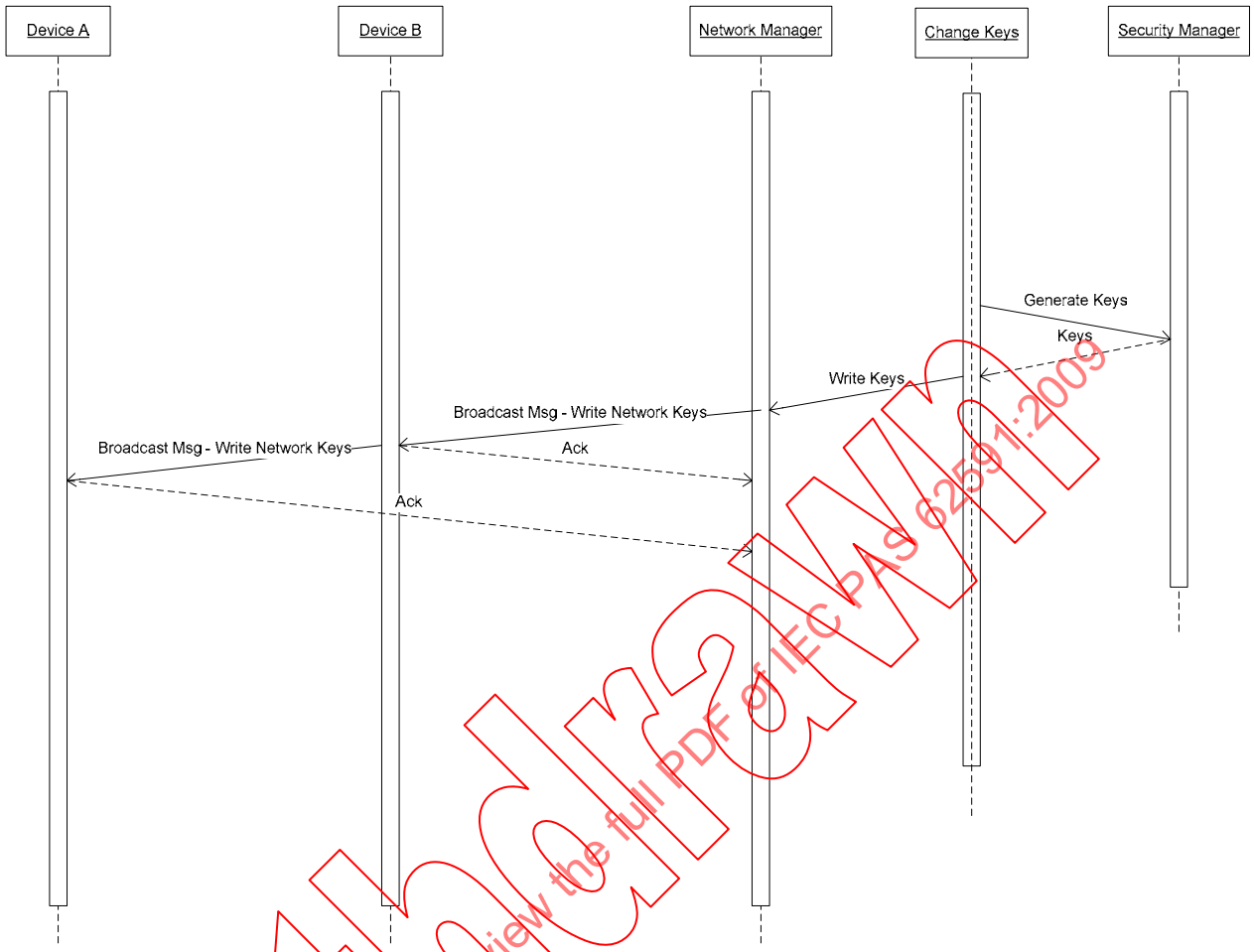


Figure 47 – Changing Network Keys

7 Wireless Devices

7.1 Purpose

Clause 7 defines requirements for specific WirelessHART device types. For compliance purposes, a WirelessHART compliant product shall be classified as one of five different device types (see Figure 21 above). These device types are:

- **Field Devices** are mounted in the process and shall be capable of routing packets on behalf of other devices. In most cases they characterize or control the Process or process equipment. A Router is a special type of field device that does not have a process sensor or control element and as such does not interface with the process itself.
- **Wireless Adapters** enable the connection of a non-native communicating field device to the WirelessHART Network.
- A **Gateway** enables communications between Host Applications and devices that are members of the WirelessHART Network. The Gateway has one or more Access Points interconnecting the Plant Automation Network and the WirelessHART Network.
- **Handhelds** and other Maintenance Tools are portable applications used to configure, maintain or control plant assets. Only portable equipment directly connecting to the WirelessHART Network falls into this category.
- The **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 WirelessHART Network.

The specification begins with an overview of the WirelessHART Network. The specification describes in detail each of the five device types summarized above.

7.2 Overview

7.2.1 Gerenal

The HART Protocol enables communication with smart process instrumentation and control devices. A WirelessHART network is made up of several components and device types. Clause 7 discusses the requirements for components of a WirelessHART Network.

7.2.2 WirelessHART Network Components

7.2.2.1 General

Figure 48 is used to discuss WirelessHART network components. In Figure 48, the network is shown connected to the Plant Automation Network through a Gateway. The Plant Automation Network could be a TCP-based network, a remote IO system, or a bus such as PROFIBUS DP. The Gateway is connected to the WirelessHART Network through Network Access Points. These Network Access Points increase the throughput and improve the overall reliability of the WirelessHART Network.

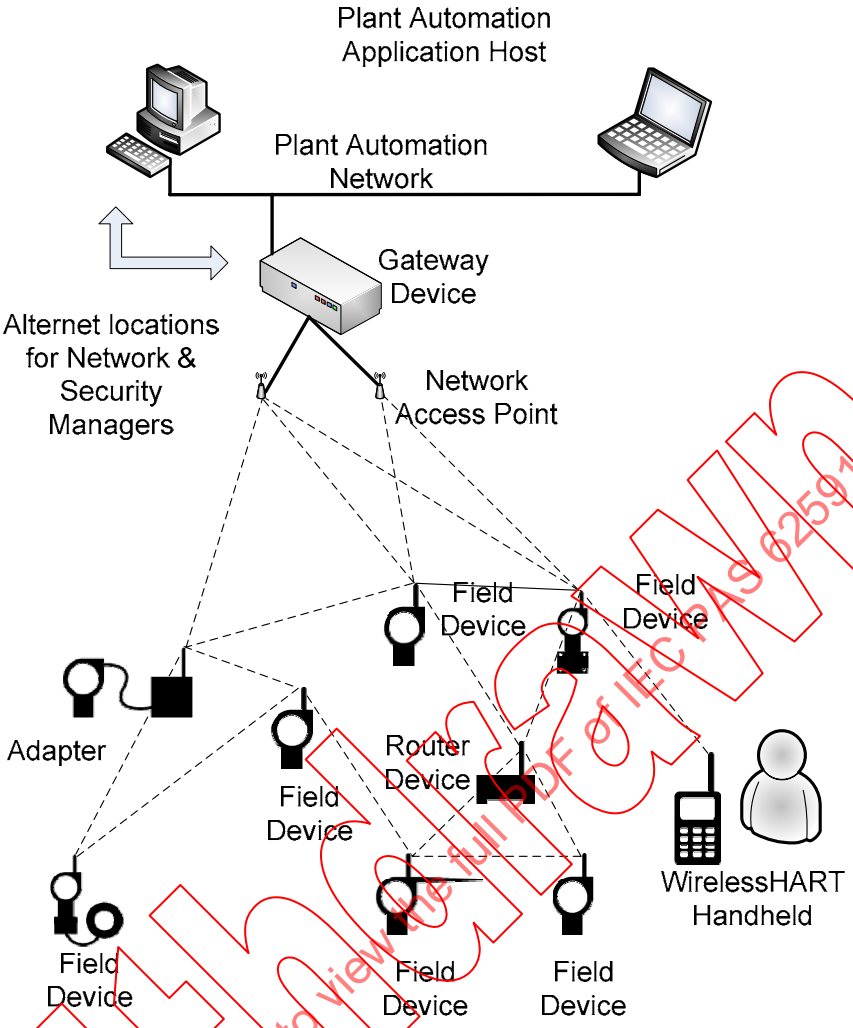


Figure 48 – WirelessHART Standalone Gateway.

All devices directly connected to the WirelessHART Network are a type of Network Device. Network Device types include Field Devices, Adaptors, Routers, Access Points and Handheld Devices.

All Network Devices transmit and receive WirelessHART packets and perform the basic functions necessary to support network formation and maintenance. All Network Devices shall be able to source and sink packets and be capable of routing packets on behalf of other devices in the network.

All Network devices have a 5 byte HART Unique ID assigned at the factory. An 8 byte IEEE address is created by appending the 5 byte HART Unique ID to the 3 byte OUI assigned to the HCF.

7.2.2.2 Field Device

Field Devices are connected to, and characterize, or, control the Process. They are a producer and consumer of WirelessHART packets and shall be capable of routing packets on behalf of other Network Devices.

7.2.2.3 Adapter

An Adapter Device is a Network Device that connects non-native communicating Devices to the WirelessHART Network. An Adapter uses internal routing tables to coordinate traffic flow

between the WirelessHART Network and its non-native communicating sub-device(s). An Adapter is not directly connected to the process.

7.2.2.4 Gateway Device

A Gateway Device is an access point that connects the WirelessHART Network to a plant automation network, allowing data to flow between the two networks. The Gateway Device provides host applications access to the Network Devices. A Gateway Device can be used to convert from one protocol to another, as go-between two or more networks that use the same protocol, or to convert commands and data from one format to another. The WirelessHART Gateway specification provides more detailed information.

The WirelessHART Network also uses the Gateway as the source for the synchronized clock used by the timeslots and Superframes.

In many situations, networks will have more than one Network Access Point. These multiple Access Points can be used to improve the effective throughput and reliability of the network. Network Access Points communicate directly with a WirelessHART Gateway, which is sometimes also referred as a Virtual Gateway. The Virtual Gateway is always the vertex of the network graph. The use of a Virtual Gateway address is discussed in the Gateway specification.

7.2.2.5 Network Access Point

A Network Access Point is a Network Device that connects Gateways into the WirelessHART Network. A Network Access Point has a WirelessHART connection on one side and an external connection on the other side – the external connection could be an Ethernet or Wi-Fi connection or a proprietary connection. The external connection is not specified by WirelessHART. A Network Access Point is not directly connected to the process. Network Access Points are discussed as part of the Gateway.

7.2.2.6 Router Device

A Router Device is a Network Device that forwards packets from one Network Device to another. A Network Device that is acting as a Router Device uses its graphs and connections to decide which Neighbor Device to send the packet. In general standalone routers are not required since all Network Devices shall support routing. However, it may be beneficial (e.g., to extend the Network, or to save the power of a Field Device in the network) to add additional devices to improve routing in the network. A router is not connected to the process and does not act as a Gateway.

7.2.2.7 Handheld Device

Handheld Devices are used in the installation, control, monitoring, and maintenance of Network Devices. Handheld Devices are portable equipment operated by the plant personnel.

There are two approaches to connect Handheld Devices:

1. WirelessHART-connected Handheld Device: A WirelessHART-connected Handheld Device communicates directly to the WirelessHART Network. When operating with a formed WirelessHART Network, this device joins the network as a WirelessHART Field Device. When operating with a target Network Device that is connected to a WirelessHART Network, the Handheld Device operates in a special mode, that mode that allows it to communicate with one device at time.

2. Plant automation network-connected Handheld Device: A plant automation network-connected Handheld Device connects to the plant automation network through some other networking technology such as Wi-Fi. This device talks to Network Devices

through the Gateway Device in the same fashion as external plant automation servers. To the WirelessHART network this type of handheld is just another host application.

7.2.2.8 Network Manager

The Network Manager is also treated as a type of Network Device. Doing so allows other HART devices to exchange HART Commands with the Network Manager.

7.3 WirelessHART Field Devices

7.3.1 Overview

The most common type of WirelessHART Network Device is a Field Device. A WirelessHART Field Device is a Network Device that combines wireless communications with traditional HART Communication field device capabilities. The Field Device may be line, loop, battery powered or powered in some other fashion. A Field Device is connected to the Process or Plant Equipment. Field Devices may or may not support traditional 4 mA to 20 mA current loop signaling. The WirelessHART field device shall have a maintenance port and may only have a wire connection to the Process Automation System.

7.3.2 General Requirements

All Field Devices shall support all HART Universal Commands. In addition, all Field Devices shall support the Commands found in Table 32.

Table 32 – Mandatory Commands for WirelessHART Field Devices

Cmd	Description
38	Reset Configuration Changed Flag
41	Perform Self Test
42	Perform Device Reset
48	Read Additional Status
54	Read Device Variable Information
59	Write Number Of Response Preambles
78	Read Aggregated Commands
79	Write Device Variable
90	Read Real-Time Clock
103	Write Publish Data Period
104	Write Publish Data Trigger

Cmd	Description
105	Read Publish Data Configuration
106	Flush Delayed Response Buffers
107	Write Publish Data Device Variables
108	Write Publish Data Command Number
109	Publish Data Control
115	Read Event Notification Summary
116	Write Event Notification Bit Mask
117	Write Event Notification Timing
118	Event Notification Control
119	Acknowledge Event Notification

7.3.3 Maintenance Port

All Field Devices shall provide a maintenance port that complies with the requirements. The maintenance port interface is used for provisioning (e.g., to load the Join Key and Network ID or to monitor the join process). All attributes and commands supported by the Field Device shall be available via the maintenance port.

NOTE Some commands are restricted (e.g., Network Manager only commands) and, consequently are not accessible via the maintenance port.

Masters or other tools connected to the maintenance port do not have access to the wireless network.

The maintenance port can be either a standard HART interface designed for connection to the Process Automation System or a dedicated maintenance port.

If it is a dedicated maintenance port, the following requirements apply:

- Through the maintenance port, the Field Device shall appear to be a HART multi-drop slave device. By default, the Field Device should be configured for polling address 1.
- The input impedance shall be 500 Ω simplifying the direct connection of legacy maintenance tools and applications supporting the HART Communication Protocol.
- The maintenance port shall not support Publish Data communication.
- The maintenance port shall not be permanently wired.
- The maintenance port shall be clearly labeled as the maintenance port.

7.3.4 WirelessHART Interface

7.3.4.1 Overview

The Field Device is a WirelessHART device and shall adhere to all WirelessHART specifications. Key Field Device requirements include:

- Support for network services; and
- Support for Publish Data operations.

Subclause 7.3.4 describes network services in the context in which they are used.

7.3.4.2 Timing requirements

All WirelessHART network devices shall be capable of routing messages on behalf of other Network Devices. Furthermore, communications shall ensure latency across the mesh is minimized and unnecessary, redundant communications is minimized. To this end, devices shall meet the following requirements:

- Be able to forward (route) a PDU (not addressed to the Network Device) in the slot immediately following the slot the PDU was received in.
- Be able to reply to a Network Management Command addressed to the Network Device in the second slot following the slot the PDU was received in (i.e., one intervening slot between the request and the response is allowed for command processing).
- Be able to reply to all other commands addressed to the Network Device in the sixth slot following the slot the PDU was received in.
- Where Delayed Responses are allowed, the DR_Initiate shall not be generated until 75 % of the Transport Layer maxReplyTime has elapsed.
- Latency shall (unless otherwise noted) represent a 2-sigma value. That means, the latency shall be achieved 95 % of the time.

7.3.4.3 Publish Data Operation

Publish Data Messages are used to publish process data to applications. In general, the Gateway provides access to the WirelessHART network and caches the published data. The Publish Data Messages are configured as required to meet the process or plant equipment requirements. This configuration is performed using standard procedures. For example, Command 108 is used to select the command to be published and Command 109 turns publishing on and off (see Figure 49 hereinafter)

Command 109 is used to start Publish Data operation and the device contacts the Network Manager to request bandwidth. Since the Network Manager may take a moment to provide the bandwidth, the Field Device initiates a Delayed Response (DR) to advise the application that

it is processing the request. The Field Device next issues an "Add Publish" service request to the Network Manager. The Network Manager returns a DR to the Field Device and begins processing the request. The Network Manager then allocates the network resources for the publish service and issues commands to set up Frames and Links. In most cases, the Network Manager will set up links in an existing Superframe to satisfy this request. Once the network has been configured, the Network Manager will return a response to the Field Device indicating that the "Add Publish" service request is complete. The Field Device will then complete the sequence by returning a response to the application indicating that command 109 is complete and Publish Data is on.

As usual, the application may issue another Command 109 to the Field Device to monitor progress on the DR. If the command is incomplete, the Field Device will respond with the status "DR_RUNNING".

Once initiated, the Field Device then publishes data indefinitely (potentially for years). The Field Device will continue to publish data until it is instructed to stop. To request a device to stop publishing data, the application issues another Command 109, this time with the command field "Off". The Field Device in-turn will send a "Delete Publish" service command request to the Network Manager to delete the service and deallocate network resources. The Network Manager will immediately answer this request and then reconfigure the network accordingly.

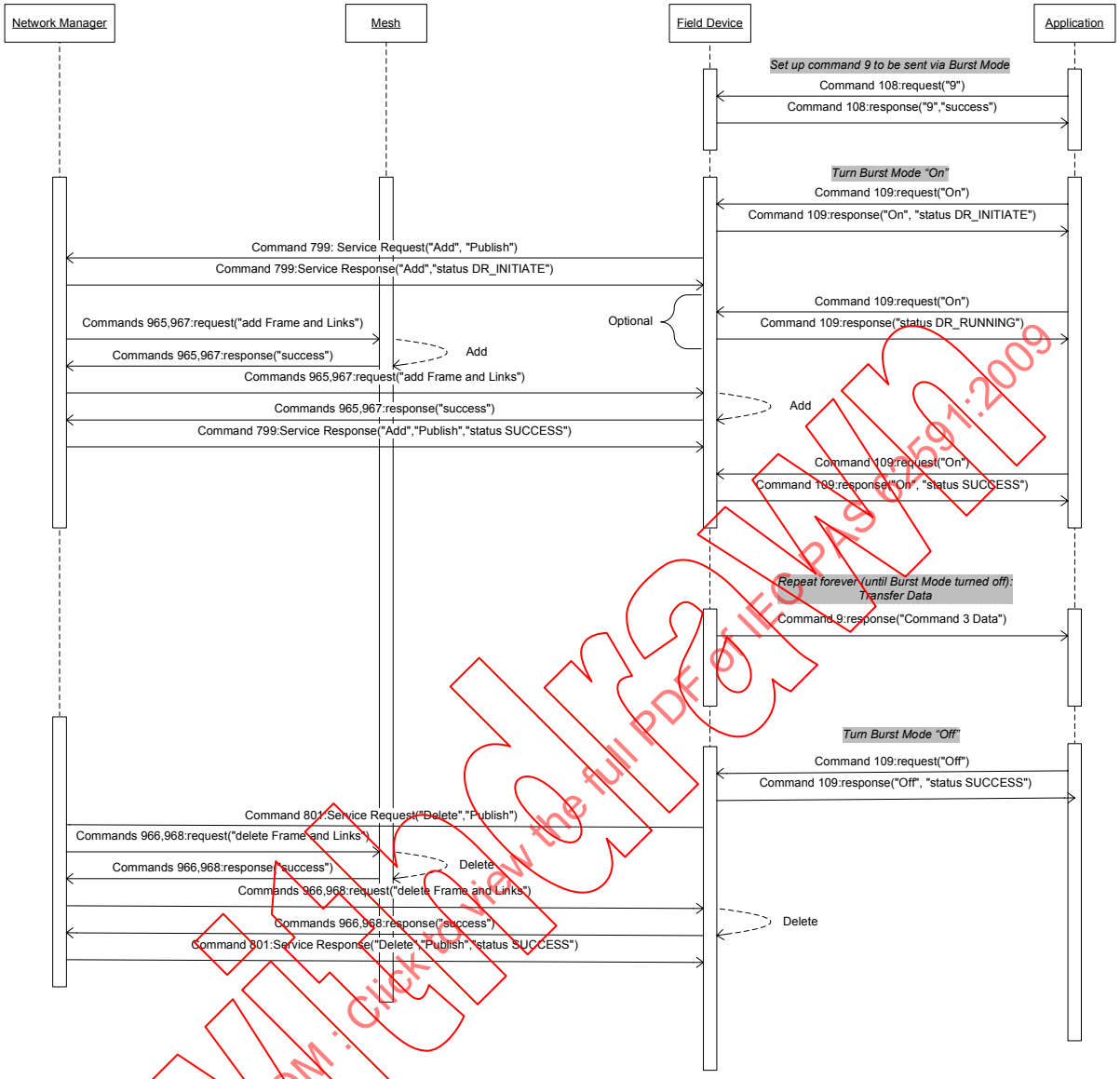


Figure 49 – Supporting Publish Data Operation

7.4 Wireless Adapter

7.4.1 Overview

A Wireless Adapter (see Figure 50) enables the connection of a non-WirelessHART native device to the WirelessHART Network. A Wireless Adapter is a WirelessHART Device supporting the WirelessHART TDMA interface. The Wireless Adapter enables communication to be passed to/from a non-native device through a WirelessHART Network. A Wireless Adapter shall meet all the requirements specified in 7.4.

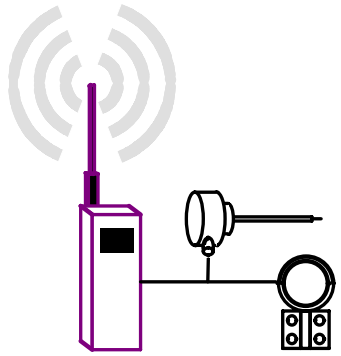


Figure 50 – Wireless Adapter

The Wireless Adapter shall support the publishing of process data on behalf of the non-native connected field device.

The Wireless Adapter shall support the Commands required of WirelessHART Field Devices and identifies itself (Manufacturer ID, Device Type Revision, Device ID, etc) in Identity Command responses. As with I/O Systems the Wireless Adapter shall set Protocol_Bridge_Device (bit 2) in the Flags byte of Identity Commands.

7.4.2 General Requirements

The Wireless Adapter builds on the requirements of WirelessHART Field Devices to support the I/O connection and tunnel communications to/from a non-WirelessHART native sub-device.

The Wireless Adapter shall act as a proxy and publish process data responses on behalf of its non-native sub-device(s). The Wireless Adapter shall support the minimum capacity requirements in Table 33.

Table 33 – Wireless Adapter Minimum Capacity Requirements

Parameter	Requirement
Minimum Number of Cards	1
Minimum Number of Channels	1
Minimum Number of Sub-devices	1
Minimum Number of Publish Data Messages	5
Minimum Number of Publish Event Messages	2

7.4.3 WirelessHART Interface

7.4.3.1 General

The Wireless Adapter is a WirelessHART device and shall adhere to all WirelessHART specifications.

7.4.3.2 Timing requirements

All Wireless Adapters shall be capable of routing messages on behalf of other Network Devices and any non-native devices it connects to the WirelessHART Network. Wireless Adapter communications shall ensure latency across the mesh is minimized and any unnecessary, redundant communications is also minimized. To this end, Wireless Adapters shall meet the following requirements:

- Be able to forward (route) a PDU (not addressed to the Network Device) in the slot immediately following the slot the PDU was received in.
- Be able to reply to a Network Management Command addressed to the Wireless Adapter in the second slot following the slot the PDU was received in (i.e., one intervening slot between the request and the response is allowed for command processing).
- Be able to reply to all other commands addressed to the Wireless Adapter in the sixth slot following the slot the PDU was received in
- Where Delayed Responses are allowed by Commands to the Wireless Adapter, the DR_Initiate shall not be generated until 75 % of the Transport Layer maxReplyTime has elapsed.

7.4.3.3 Publish Sub-Device Process Data

The Wireless Adapter shall Publish Data on behalf of its connected non-native sub-device(s). The Publish Data Messages are configured in the Wireless Adapter as required to meet the process or plant equipment requirements. The Wireless Adapter is responsible for acquiring the data (as needed) from the sub-device. The Wireless Adapter complies with the Publish Data requirements and generates the data on the specified schedule.

7.5 WirelessHART Gateway

7.5.1 Overview to this subclause

Subclause 7.5 describes the WirelessHART Gateway. The Gateway is functionally divided into a Virtual Gateway and one or more Access Points. Multiple Access Points increase the throughput and the reliability of a WirelessHART Network. To simplify support for redundant Access Points, every Gateway has a fixed, well known address (Unique ID = 0xF981 0x000002; Nickname = 0xF981). There is one Gateway per network. In addition, each Access Point has a Unique ID, EUI-64 address. The Nickname (short address) for the Access Point is assigned by the Network Manager. The scope of the Gateway is shown in Figure 51.

By de-composing a Gateway into a Virtual Gateway and one or more Access Points allows the Gateway reside at the root of all graphs. Consequently, packets can be routed to the most convenient Access Point and, if an Access Point fails, packets shall flow to the remaining Access Points. Network traffic will be constrained but communication will still be successful.

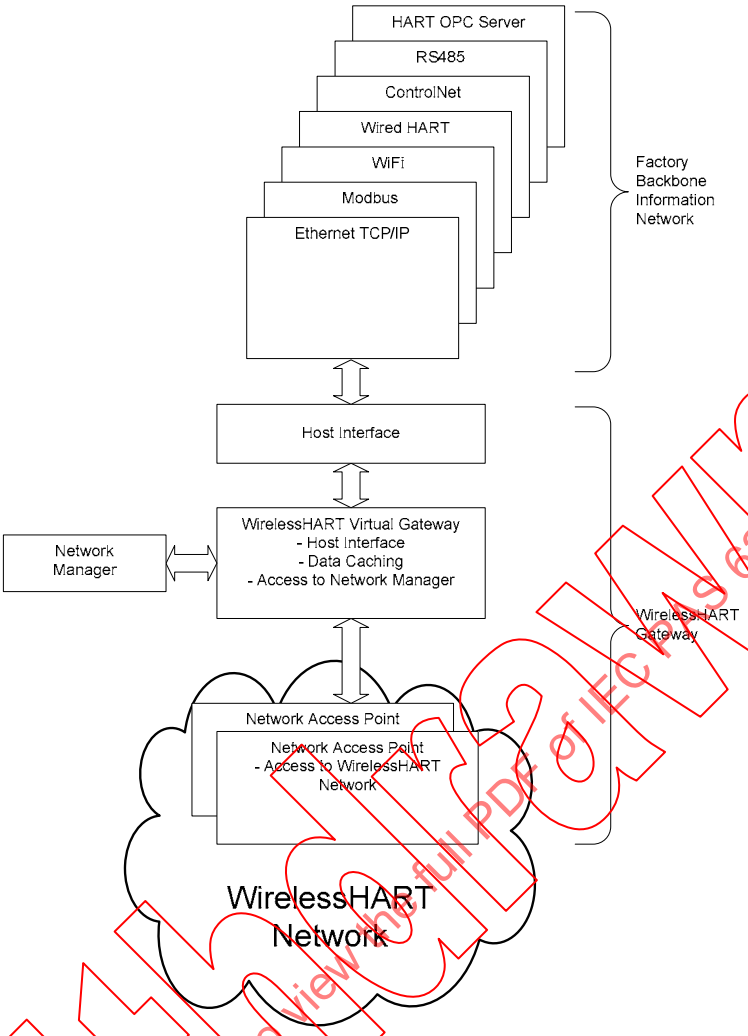


Figure 51 – Gateway Scope

7.5.2 General Requirements

A WirelessHART Gateway is subdivided into a Virtual Gateway, one or more WirelessHART Network Access Points, and one or more Host interfaces. The WirelessHART Gateway provides the following:

- one or more Access Points providing the physical connection into the WirelessHART Network;
- a Virtual Gateway providing a sink or source point for WirelessHART Network traffic;
- one or more Host Interfaces connecting the Gateway to backbone networks (e.g., the plant automation network;
- a connection to the Network Manager;
- buffering and local storage for Publish Data, event notification, and common commands (e.g. Commands 0, 20, 48);
- time synchronization sourcing;
- support for WirelessHART Adapters; and
- backward compatibility with legacy applications.

The Gateway uses standard HART commands to communicate with network devices and host applications. The Gateway also acts as a server responsible for collecting and maintaining cached data and command responses from all devices in the network. These cached responses correspond to Publish Data Messages, event notifications, and common HART

command responses. These cached responses are returned immediately to host application requests. This reduces network communication load improving power utilization and host application responsiveness.

If multiple Access Points are supplied by the Gateway, the Network Manager will schedule communication traffic through all of them. If one of these Network Access Points fails, then the Network Manager will adjust the schedule spreading traffic across the remaining Network Access Points. Each Access Point has its own physical and nickname Address.

Internal to the Gateway, all Access Points route traffic through the Virtual Gateway (see Figure 52) with a Host Interface or the Network Manager.

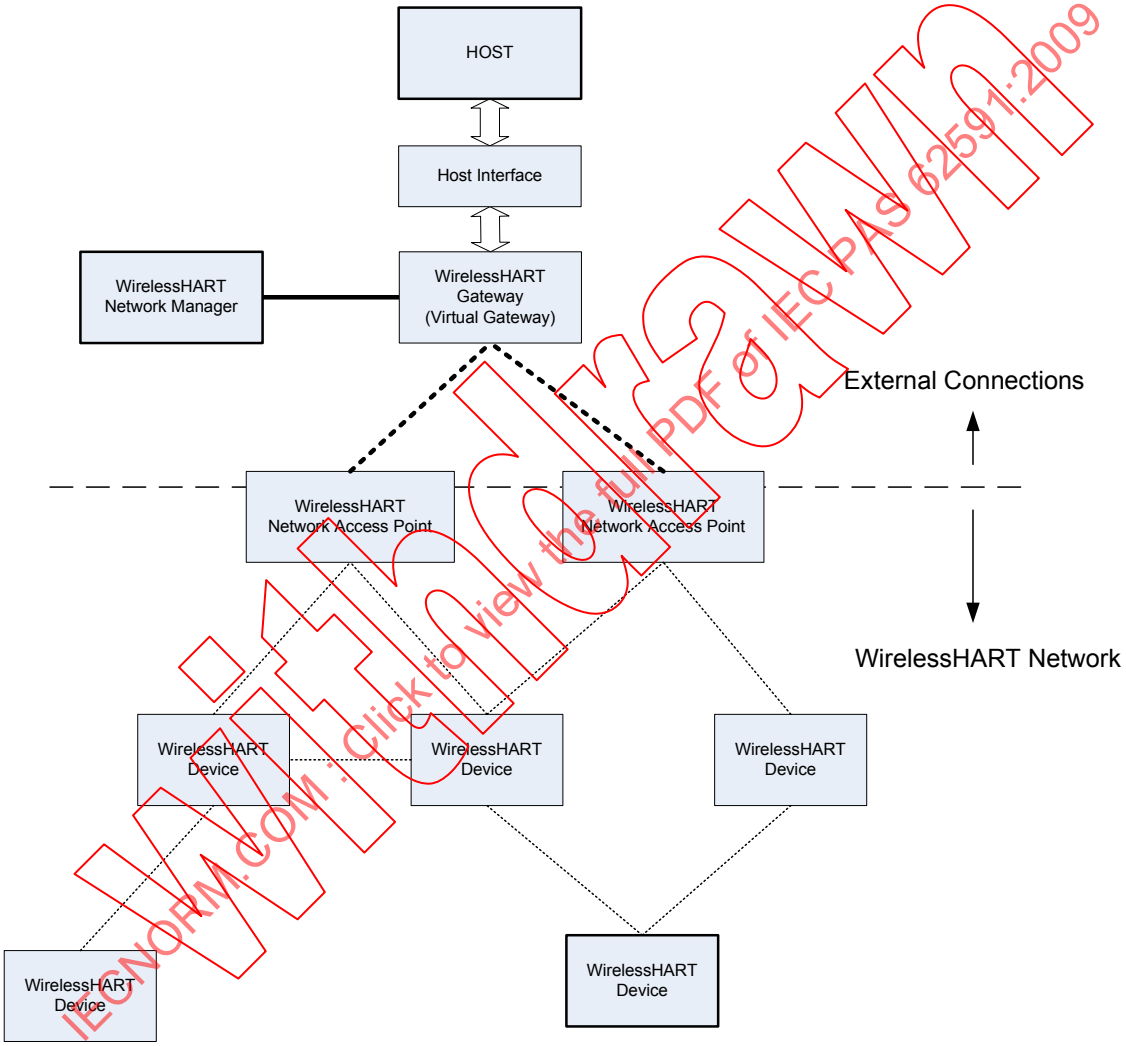


Figure 52 – Virtual Gateway and Network Access Points in a WirelessHART Network

The WirelessHART Gateway shall provide the network clock to other Network Devices. The clock information ripples downward from the top of the network hierarchy to the bottom.

The WirelessHART Gateway shall support Commands and identifies itself (Manufacturer ID, Device Type Revision, Device ID, etc.) in Identity Command responses. In addition, the WirelessHART Gateway supports a number of Gateway Specific Commands.

7.5.3 Gateway Model

7.5.3.1 General

The WirelessHART Gateway has several distinct components as shown in Figure 53. The Virtual Gateway itself is a type of Network Device. The Virtual Gateway makes use of services from the Network Manager and the Security Manager to authenticate and join network devices.

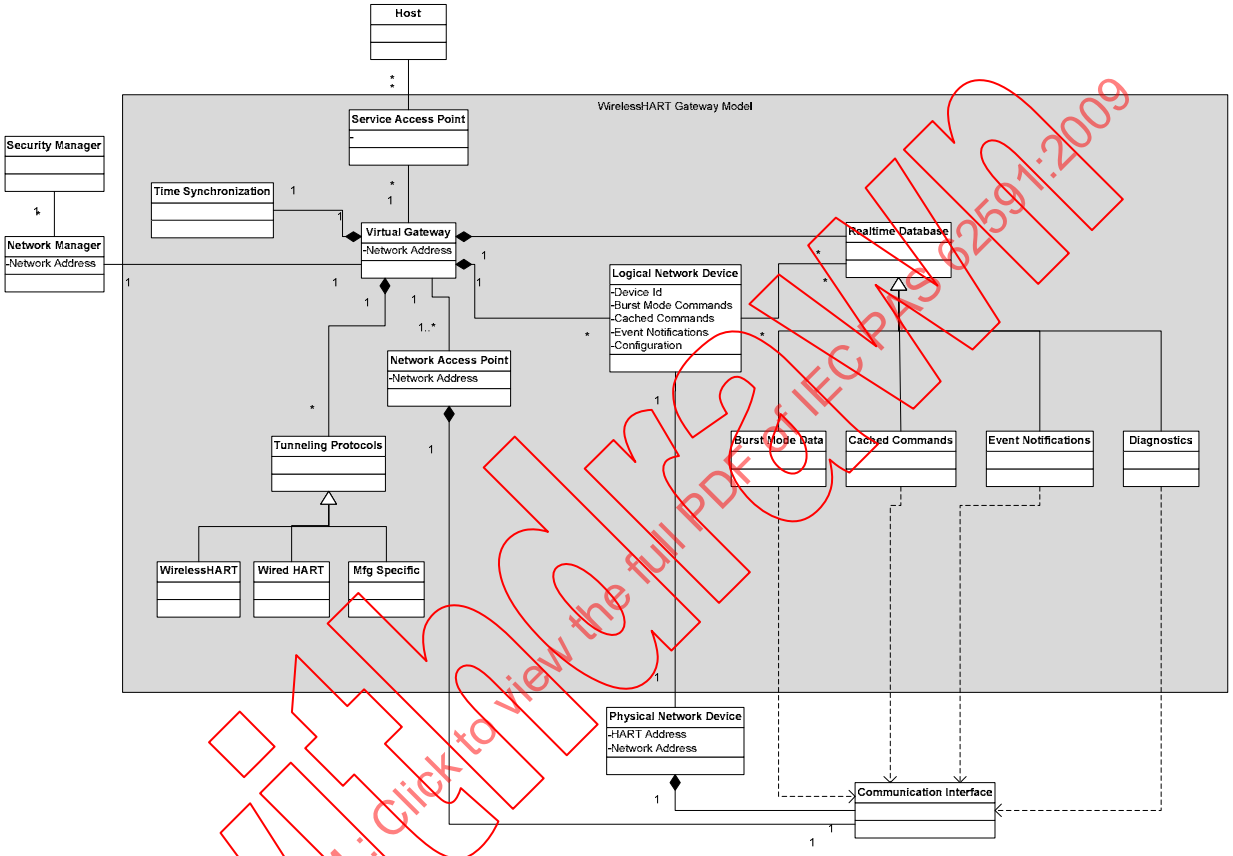


Figure 53 – Gateway model

WirelessHART Gateways connect the WirelessHART Network with other networks, such as plant automation networks, allowing HART Commands/Responses, tunneled messages, formatted XML, and diagnostic messages to flow between the two networks. The WirelessHART Network uses the concept of a Virtual Gateway to provide a single entry point into the WirelessHART Network. Access to Host interfaces is through Service Access Points. Access to the WirelessHART Network itself is provided through Network Access Points. Each of the items identified in the drawing above is described in more detail in the following subdivisions of 7.5.3.

7.5.3.2 Virtual Gateway

The Virtual Gateway provides a single entry point into the WirelessHART Network.

- It is part of the WirelessHART Field Device network.
    a. It is a device type in the WirelessHART Network.

    b. It communicates through Access Points to any Field Device in the WirelessHART Network (the Virtual Gateway shall have a path to every device in the WirelessHART network).

- It can communicate directly with the Network Manager.
- It sources time synchronization messages.
- It is a HART Device Type that supports and is described by EDDL, see IEC 61804-3.
- It supports one or more Service Access Points for connecting to the automation network and plant backbone. It supports the following through these Service Access Points:
    a. Translation functions satisfying HART Commands with locally cached data. The WirelessHART Gateway implements a data cache to optimize the overall performance of the WirelessHART network and improve responsiveness to host applications.

    b. Tunneling functions transferring HART Commands to WirelessHART Network requests. The WirelessHART Gateway can connect with the host application via various protocols (e.g., Modbus, Profibus DP, ControlNet, HART OPC server, proprietary, other), based on different physical layers (RS-485, Ethernet LAN, Wi-Fi, etc.).

    c. Optionally supports an XML-based interface.

- Compatibility
    a. The WirelessHART Gateway can support existing HART commands (only to the extent that the Gateway is acting as a translator or proxy).

- It provides buffering for
    a. Publish Data Messages.

    b. Event Notification.

    c. Cached command responses.

    d. Diagnostics.

    e. Large data transfers (several specific cases have been discussed, for example a valve uploading its signature information and the results from a vibration analysis are two use cases where the gateway is receiving from a device; the gateway can also perform large data/file transfers down to a device).

- It provides support for publishing variables to devices (often referred to as catch variables). In this case the WirelessHART Gateway will be able to publish device variable data that it is caching to other devices in the WirelessHART network.

The network used on the host side may consist of a variety of technologies. Most PLC, DCS or SCADA vendors utilize a proprietary network. Asset Management and Device Management companies tend to use open protocols, such as TCP/IP and one of several standard MAC/PHY layers such as IEEE 802.11 and IEEE 802.3.

7.5.3.3 Access Point

Network Access Points provide access to the WirelessHART Network. They provide the following:

- They are part of the WirelessHART Field Device network.
    a. They are a device type in the WirelessHART Network.

    b. They communicate with the Virtual Gateway via dedicated link or communication port.

    c. Each Network Access Point can support communication with any device to which the Network Manager has provided a path.

7.5.3.4 Service Access Point (SAP)

7.5.3.4.1 General

Service Access Points provide a connection to the automation network and plant backbone. They provide:

- An interface to the Virtual Gateway for host systems or applications that wish to access Network Devices that are part of the WirelessHART Network. The interface provides support for accessing all wired HART Devices that are included through Adapters.
- Access to cached response messages:
    a. Publish Data Reponses.
    b. Event Notification Reponses.
    c. Cached command responses.
- Access to diagnostics.
- Access to Network Manager data.
- Support for block mode data transfers (e.g. uploading results from a vibration analysis).
- Tunneling functions transferring HART Commands to WirelessHART Network and WirelessHART Device requests. Through these SAPs the Virtual Gateway can connect with the host application via various protocols (e.g.[3] Modbus, Profibus DP, ControlNet, HART OPC server, proprietary, other), based on different physical layers (RS-485, Ethernet LAN, Wi-Fi, etc.).

To support Service Access Points, two interface types are provided. The first directly supports HART Commands – this interface shall be supported by all gateway implementations. The second supports XML-formatted commands – the XML interface is optional.

7.5.3.4.2 WirelessHART Interface

A WirelessHART Gateway shall be able to tunnel HART commands to/from any WirelessHART device.

7.5.3.4.3 Vendor Specific Proprietary Protocols

A WirelessHART Gateway supporting open and vendor specific proprietary protocols shall be able to tunnel HART command request/responses through the open and proprietary protocols. For example, most PLC, DCS or SCADA vendors utilize a proprietary network. Vendors of Asset Management and Device Management applications tend to use open protocols, such as TCP/IP and one of several standard MAC/PHY including 802.11 and 802.3.

7.5.3.5 Tunneling Protocols

Gateways shall also be able to support tunneling protocols. Tunneling protocols are used to relay messages between the host which is outside the WirelessHART Network and a destination device that is part of the WirelessHART Network. All WirelessHART Gateways shall be able to support HART and WirelessHART commands. They may also support manufacturer specific commands.

There are several categories of Tunneling Gateways – WirelessHART, HART over Ethernet, open protocol such as TCP/IP, and Vendor Specific. Examples are described below.

[3] Modbus, Profibus DP, ControlNet, HART OPC server are examples of suitable products available commercially. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of these products.

7.5.3.6 Host Interface

The Gateway Host interface is used to connect client applications outside of the WirelessHART Network with the WirelessHART Network and the devices in the WirelessHART Network. The Gateway Host interface can take many forms. Several common ones include the following:

- Ethernet-to-wireless Gateway Device: A Gateway Device that provides a bidirectional path between industrial Ethernet Networks and the WirelessHART Network.
- Wi-Fi-to-wireless Gateway Device:A variation of an Ethernet-to-wireless Gateway Device that uses 802.11 a/b/g radio to connect to the plant's network.
- Serial-to-wireless Gateway Device: If plant automation servers and equipment support serial interfaces, a serial-to-wireless Gateway Device can be used to connect to the serial interfaces of these devices.

The WirelessHART Gateway shall be able to cache Publish Data command responses, commonly used to read and write commands, and diagnostics data from the WirelessHART Devices and Wireless Adapters in the Network. To take advantage of this caching, the Gateway shall also be able to act as a translator. As a translator, the Gateway peeks at requests from the Host applications and, if the response data is cached and is current, returns the cached response messages from its real-time database. For example, if a host issues HART command #0 request to a device in the network, the Gateway will check to see if it has a cached command #0 response. If the Gateway does not have a cached command #0 response for that device, it will forward the command to the device and return the resulting response to the client.

The translation functions can be quite involved. They deal with network layer as well as some application layer interactions. In the network layer, the different response packet sizes have to be dealt with, and a mapping of security, priority, addresses and such is made.

7.5.3.7 Logical Network Device

The Gateway maintains a list of Logical Network Devices. The device information cached in the Gateways in turn contains information about the devices such as the list of Publish Data Commands, Event Notification Messages, and Cached Commands Responses that are currently stored there. The Logical Network Device is shown in Figure 54.

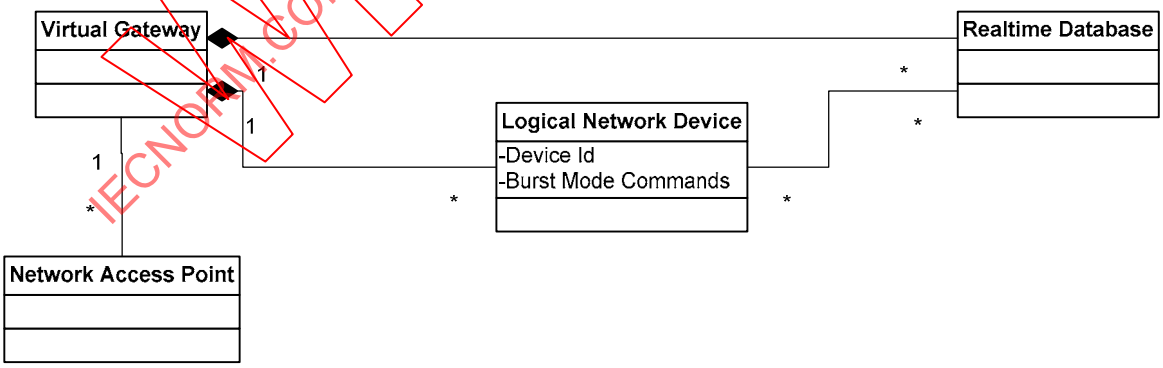


Figure 54 – Logical Network Device

The Logical Network Device plays an important role in modeling the system and later in commissioning the system. A Logical Network Device can exist independent of an actual Physical Network Device. The Logical Network Device provides a network placeholder that can be used for off-line configuration, simulation, and on-line operation. It also provides the necessary separation for commissioning and device replacement.

7.5.3.8 Physical Network Device

A Physical Network Device is an actual device in the network. A Physical Network Device discovers neighbors and builds and maintains a neighbor table. This neighbor table is built by the physical device by listening for a specified amount of time on each channel. Each neighbor is recorded with its corresponding received signal strength.

The Physical Network Device is shown below in Figure 55.

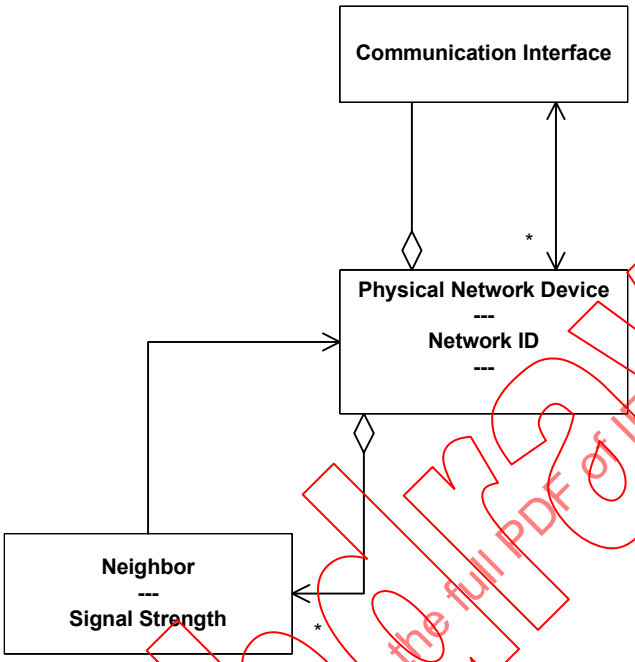


Figure 55 – Physical Network Device

7.5.3.9 Communication between Gateway and Network Manager

The WirelessHART specification does not specify the architecture of the Network Manager, Gateway, or Security Manager. The Network Manager and Gateway may be physically combined or separate.

In all designs, the Network Manager and the Gateway shall establish and maintain a secure communication channel with each other. All communications with the WirelessHART Network pass through the Gateway. Consequently, the Gateway shall route packets to the specified destination (Network Device, host application, or Network Manager).

The Network Manager creates an initial Superframe, assigns links in it for the Gateway's Access Points, and configures the Gateway. ASN 0 is established when this initial frame is activated.

The Network Manager is not involved in communications between host applications and network devices. The Gateway is responsible for buffering, protocol conversions, timeouts, and time synchronization.

7.5.3.10 Communication Interface

7.5.3.10.1 General

The Communication Interface provides the required communication services to talk with Network Devices. In the implementation, these communication services will be provided by the Network and Transport layers. The service types are summarized below:

- Request / Response;
- Block Data Transfers;
- Publish Data Messages;
- Event Notification;
- Diagnostic Service Type.

7.5.3.10.2 Request/Response

The Request/Response Service provides a method for the Network Manager and Host Applications to send request messages to Network Devices and to receive their response messages. If the response takes too long, the Gateway will send either a DR or a "timeout" response to the Host application. In all cases, the application expects a confirmed HART response message from the Network Device.

To the Network Manager these communications are "unscheduled" and occur ad-hoc. Consequently, a base of bandwidth shall be available to accommodate request/response traffic. The Gateway is responsible requesting this service, increasing and decreasing the bandwidth it requests as application demands vary.

7.5.3.10.3 Block Data Transfers

Block data transfers shall also be supported both to Network Devices and through an Adapter to its subdevice. This supports transmission of large data sets with automatic segmentation and re-assembly the destination.

7.5.3.10.4 Publish Data Messages

The Publish Data Messages are used to send data on a periodic schedule or on an exception basis. A Publish Data Message consists of the response packet for the specified command. The Publish Data rate is configurable, based on application needs.

For control applications, the Publish Data rate is determined by control loop or sequence execution requirements (e.g., the process time constant). In some cases, more than one host application will subscribe to the same published data. In these cases, the update rate is determined by the fastest requested data rate.

Publish Data responses shall be cached inside the Gateway in a real-time database. This provides a mechanism for multiple readers to access the same data, decouples the applications and reduces duplicate network traffic.

7.5.3.10.5 Event Notification Services

The Event Notification Services provide support for the communications of events and alarms. This is a guaranteed message delivery service. The Gateway is responsible for acknowledging and caching Event Notification Messages.

7.5.3.10.6 Diagnostic Service Type

This service is used to transfer diagnostic information about the network to the Host.

7.5.3.11 Cached Response Messages

7.5.3.11.1 Network Status

Each Network Device maintains diagnostics. The diagnostics are periodically published via HART commands to the Network Manager. The Network Manager maintains the complete set of Device and Network Diagnostics. Hosts can query the Gateway or the Network Manager for network level diagnostics.

7.5.3.11.2 Publish Data Command Responses

The database caches all of the Publish Data response messages.

7.5.3.11.3 Event Notification Command Responses

The database caches all of the event notification response messages.

7.5.3.11.4 Cached Command Responses

The database caches the latest response messages for several commands (these commands are summarized below).

7.5.3.11.5 Delayed Response Command Responses

For HART request/response commands, the Gateway maintains a complete list of all outstanding commands that have been sent to devices for which a response has not yet been received in return. Delayed Response Commands shall be purged if they exceed a 24 h timeout.

7.5.4 Gateway Management

7.5.4.1 Addressing

Between the WirelessHART Gateway and the Host, each Field Device is identified by its 5-byte HART address. In addition, for Network Devices, the Network Manager maintains a unique 2-byte address (i.e. Nickname) for each device in the network. Some devices, for example those connected through an Wireless Adapter, do not have a Nickname. In all cases the Gateway maintains a table of 5-byte HART addresses, Nicknames, and device type information. This table is used by the Gateway to translate addresses between Host (or Clients) and the network.

7.5.4.2 Retry Mechanisms

Result of the host application request to the Gateway may be classified as follows:

- Success: A valid response message is generated within the prescribed timeout window.
- Busy: The Gateway received the request but is unable to respond at the present time.
- DR: The Gateway received the request and has started processing it.
- Error: The Gateway received the message but detected an error in it that prevents it from being processed.

The number of times that a Gateway will retry before the Gateway returns to Busy or an Error, can be configured.

7.5.4.3 Power-on Reset

The WirelessHART Virtual Gateway will perform the following sequence when it is powered on:

- Calculate the power outage time. If the power outage time is less than 15 min, set an internal WARM_START flag. If the power outage time is greater than 15 min, set the COLD_START flag. If power outage time cannot be determined, set the COLD_START flag.
- Look for Network Access Points and form connections with them.
- If a Network Manager is found with the same Network Id, connect to that Network Manager.
- Synchronize the Gateway's clock with an external time source such as a GPS receiver. The Gateway will synchronize it's clock with the external time source and, at least once per hour, broadcast a UTC Time Messages.

NOTE This is a separate mechanism from keep-alives which are used to keep the networks understanding of the ASN in-sync.

- Initialize host interfaces. The Gateway can now begin returning information to host applications.
- If WARM_START
  - Discover the networks Absolute Slot Number (ASN).
  - Check connection to all network devices in the graph table by sending commands 0 and 20 to each of the devices.
  - For each Adapter execute the "Join Sequence for Adapter Sub-Devices". Update the devices status to match the response returned from command 75 "Poll Sub-Device".
- Commence normal operations.

7.5.4.4 Network Access Point Reset (HART Command #42 sent to a Network Access Point)

When a Network Access Point is reset it will completely clear its memory and restart. When it starts up it will look for the Virtual Gateway and form a connection. Once it has found the Virtual Gateway it will join the network through the Virtual Gateway and then begin looking for neighbors on the WirelessHART side.

7.5.4.5 Gateway Reset (HART Command #42 sent to the Virtual Gateway)

Resetting the Gateway is treated the same as COLD_START. The WirelessHART Gateway will set its internal Gateway RESET flag, clear all of its buffers, and tell the Network Manager that it is being reset. The RESET flag will not be cleared until network communications have been re-established. While the reset sequence is underway, the WirelessHART Gateway will respond to all commands to field devices with Busy (code #32). Any existing delayed responses and block mode transfers will be cleared and no new ones will be accepted until the reset sequence has been completed. The following summarizes the actions taken by the WirelessHART Gateway:

- Set RESET flag.
- Fail any outstanding Block Mode Transfers.
- Fail any outstanding Delayed Responses.
- Send a command '03xxx Reset Device' to the Network Manager (tell the Network Manager that the Gateway is being reset).
- Clear all tables and buffers.
- Initialize host interfaces.
- Re-connect with the Network Manager.
- Get re-configured by the Network Manager.
- Send Command #0 and Command #20 to each device.
- For each Adapter execute the "Join Sequence for Adapter Sub-Devices".

- Re-establish periodic update messages with all devices.
- Clear RESET flag.

7.5.4.6 Re-build Publish Data Periodic Data

When a WirelessHART Gateway's REBUILD flag is set, all cached responses for a specified device are cleared. In the case of a Wireless Adaptor, this means that all buffers for all devices connected to the Wireless Adapter will also be reset. The WirelessHART Gateway will use command #0 to test the connection to any WirelessHART device.

7.5.4.7 WirelessHART Gateway Self Test (Command #41)

The WirelessHART Gateway may do the following after accepting command #41:

- verify the ROM checksum (and set a flag if the check fails);
- verify the non-volatile memory contents (and set a flag if the check fails);
- generate a command #41 response message.

These ROM check and nonvolatile memory verification functions may be performed periodically by the WirelessHART Gateway.

7.5.4.8 Adding New Network Devices

Whenever the Gateway receives a new Network Device in its Route Table (i.e., the device is received from the Network Manager), the Gateway sends the device a Command 0 and Command 20. The response messages are cached for that device in the Gateway. The Gateway determines which devices are Wireless Adapters (i.e. a protocol bridge device), by examining byte 8 (Flags), bit 2 (Protocol Bridge Device) of the Identity Command (Command 0).

7.5.4.9 Device Configuration Change Status Notifications

Whenever the Gateway receives a Configuration Change Status, it shall send a Command 0 and a Command 20 to the device. The response messages are then cached for that device in the Gateway. The configuration change counter is stored as part of the Command 0 response. If the device is a Wireless Adapter, then the Gateway repeats the "Join Sequence for Adapter Sub-devices" connected to the Wireless Adapter.

7.5.5 WirelessHART Gateway Superframe

The Gateway communicates with the Network Devices via its Access Points. The Access Point should have activity (e.g., a transmit or receive) scheduled for every slot. Not utilizing every slot represents wasted opportunities. For example, if the access points have nothing else to do, they should advertise and perform shared listens.

The Network Manager should assign unused Access Points to Advertise faster than ChannelSearchTime. In doing so, Devices trying to join will quickly identify the Access Point (if it is in range) and join.

Generally, a dedicated superframe should be assigned by the Network Manager (e.g., superframe number 253). By allocating a high-numbered Superframe ID, other transmit and receive links can be used to transmit or receive higher priority traffic (in fact every other transmit/receive is more important).

7.5.6 Gateway Change Notification Services

The Gateway can return change notification messages to the Host (Client) when changes are detected by the Gateway. These notifications provide an indication that a value or status has

changed – they do not include the actual changed values. When a Client receives a change notification, it can issue a Request message to the Gateway to read the associated information. For example, when a Client receives a change notification from the Gateway for a Publish Data update from a device, the Client could issue a request to the Gateway to return the cached Response message.

NOTE Not all host interfaces can support change notifications.

The changes for which a Client can receive change notifications on are summarized in Table 34.

Table 34 – Required Command Responses

Notification Type	Fastest Notification Rate (s)	Comments
Publish Data	0,250	The Gateway checks the cache on a device-by-device basis for Publish Data changes (e.g., cmd 9). If there are changes, a change notification is added to the notification list
EventNotification	1	Every 1 s the Gateway checks the cache on a device-by-device basis for event notification updates. If there are changes, a change notification is added to the notification list
DeviceStatus	5	Every 5 s the Gateway checks the cache on a device-by-device basis for device status changes (monitors device communication status changes). If there are changes, a change notification is added to the notification list NOTE Since the device status is always sent with the Publish Data message, when Publish Data is enabled this rate will be increased up to the Publish Data rate
DeviceConfiguration	60	Every 60 s the Gateway checks the cache on a device-by-device basis for device configuration changes (monitors command #0 and command #20, checks configuration change bit and configuration change counter). If there are changes a change notification is added to the notification list. NOTE Since this can also be handled with each Publish Data, the notification should only be sent if no Publish Data message has been sent
NetworkTopology	60	Every 60 s the Gateway checks with the Network Manager to see if there have been network topology changes. If there are changes, a change notification is added to the notification list
NetworkSchedule	60	Every 60 s the Gateway checks with the Network Manager to see if there have been network schedule changes. If there are changes, a change notification is added to the notification list

The Gateway shall support at least 8 (32 recommended) separate change notification lists (i.e. 8 clients can register for change notification messages). These change notification lists are for host-side support.

The following diagram illustrates the sequence a Client could use to request the Gateway to monitor several devices. The overall sequence occurs in the following order:

1. Client1 requests change notification messages for three devices, Device1, Device2, and Device3. The client does this by issuing HART Command 140 to the Gateway.

2. The Gateway grants the request and sets up a change notification service for Client1. The Gateway returns a HART Response message indicating that the change notification request was successful.

3. Device2 sends a command 3 using Publish Data update. The Gateway caches the response message for command 3 for Device2 and sets the change notification bits for command 3 on Device2.

4. The Gateway processes the change notification list for Client1. It adds command 3 for Device 2 to its change list, sends a HART Command 142 to Client1 indicating the changes that have occurred. It then clears the change notification bits for Client1.

5. Client1 receives the change notification message from the Gateway and reads the cached command 3 Response message from the Gateway's cache.

The sequence diagram is illustrated in Figure 56.

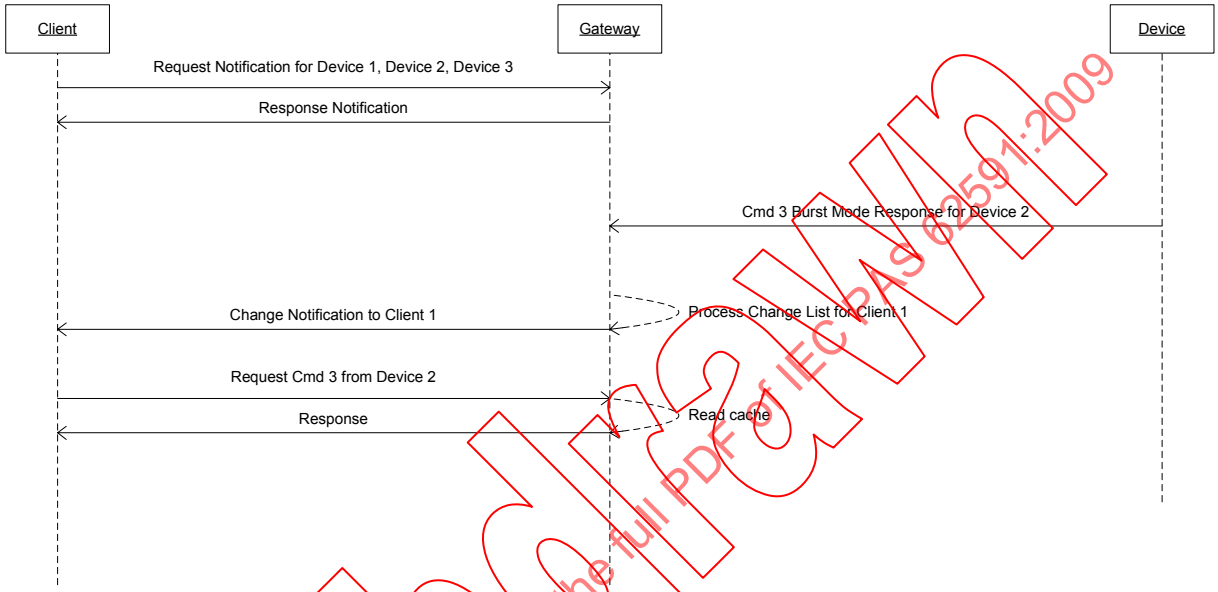


Figure 56 – Managing Notification Services

7.5.7 HART Commands Interface

7.5.7.1 General Requirements

The WirelessHART Gateway provides an interface between Host applications and the WirelessHART Network. The WirelessHART Gateway will:

- Always use extended HART addressing (except command #0).
- Cache Publish Data response message data and device status data.
- Cache selected HART read/write messages.
- Cache Event Notification response messages.
- Support client-side service requests for request/response, change notification, Publish Data, block mode transfers, delayed response messages, and high throughput services.
- Pass through command requests and responses addressed to/from devices in the WirelessHART network.
- Automatically request increases and decreases in communication bandwidth between the Gateway and field devices in response to host application demands.

Like other HART-enabled devices, the Gateway shall retain its configuration across resets and power failure. Network schedule (superframe, links, routes, etc., are not retained). These parameters include (but are not limited to):

- HART Address and Nickname per Access Point;

- Network Id;
- Join Key;
- Tag, Descriptor and Date;
- Retry and timeout limits for busy and other errors.

7.5.7.2 Host to WirelessHART Command Request and Response

The following commands are commands from a host that are addressed to the WirelessHART Gateway itself:

- Command #0, Read Unique ID, will be recognized in short or extended format provided the address matches that of the WirelessHART Gateway in either case.
- Command #11, Read Unique ID associated with Tag, and Command #21, Read Unique ID associated with Long Tag, will be recognized only if the tag and address match those of the WirelessHART Gateway.
- All other commands shall be in extended 5-byte addressing format (and match).
- If a communications error is detected, the WirelessHART Gateway will respond with a comms error response.
- The WirelessHART Gateway will respond with Command Not Implemented (error code #64) if a command number is unknown.
- On a RESET, the WirelessHART Gateway will respond with Busy (error code #32) until the RESET flag is cleared.

7.5.7.3 Pass-through of HART Command Request and Responses

Pass-through service allows a HART command request from a host to a specific field device to pass-through the WirelessHART Gateway and then through the WirelessHART network to the destination device and for the command response to return through the network to the Gateway and pass-thru to the host.

- All commands shall be in extended 5-byte addressing format.
- If the command matches one in the Gateway cache, the WirelessHART Gateway will return the cached Response message.
- If the response is not cached, the WirelessHART Gateway will pass-thru the request to the field device and, and on return, cache the response (if it is one of the commands that it supposed to cache) and return the response from the field device to the host.
- The host can flush completed delayed responses (command #106).
- If a communications error is detected, the WirelessHART Gateway will respond to the host on the devices behalf with the appropriate comms error response.

7.5.7.4 Caching Publish Data Command Response Messages

The dynamic data cache in the WirelessHART Gateway updates and records each time a device sends Publish Data message response including the data validity flags, byte count, response bytes, data bytes, total numbers of updates and the extended status bytes.

7.5.7.5 WirelessHART Gateway Status Error Flag Bits

The WirelessHART Gateway supports the Status Error Flag Bits shown in Table 35.

Table 35 – WirelessHART Gateway Status Flags

Error	Description
Cold-start	Set when the WirelessHART Gateway is powered-up or reset and the amount of time the Gateway has been unavailable is more than 15 min; cleared when the Gateway completes its start-up sequence
Warm-start	set when the WirelessHART Gateway is powered-up and the amount of time the Gateway has been unavailable is less than 15 minutes, cleared when the Gateway completes its start-up sequence NOTE – this is an internal state in the Virtual Gateway.
Configuration-changed	Set whenever WirelessHART Gateway parameters are changed by the host (any write / reset), cleared by command #38
Malfunction	OR of HARD_FAULT bits (see command #48 and variable def.)
More status available	Command 48 should be read

7.5.7.6 WirelessHART Gateway Additional Status Flags

In responding to command #48, the WirelessHART Gateway will only report the current state of its various flags (note use command #41 to have the Gateway run a self-test).

7.5.7.7 WirelessHART Gateway Capacities

The Gateway shall also provide the minimum capacity to support devices as indicated in Table 36. The minimum capacities are summarized in Table 36.

Table 36 – Gateway Minimum Capacity Requirements

Parameter	Tiny Gateway (10 devices)	Small Gateway (50 devices)	Large Gateway (250 devices)
Minimum Number of Sessions	30	110	510
Minimum Number of Transport	30	110	510
Minimum Number of Route	15	60	128
Minimum Number of Neighbors	12	50	128
Minimum Number of Superframes	12	12	12
Minimum Number of Links	50	100	500
Minimum Number of Graphs	25	60	128
Minimum Number of Graph-Neighbor	40	200	1 000
Minimum Number of Packet Buffers	25	100	500
Minimum Number of Publish Data Messages	30	200	1 000
Minimum Number of Cached Messages	75	400	2 000
Minimum Number of Publish Event Messages	25	100	500
Total Number of Devices (Wireless Adapters + Field Devices)	10	50	250
Clients	4	8	32

7.5.7.8 WirelessHART Gateway Commands

7.5.7.8.1 General

The WirelessHART Gateway supports several Gateway specific commands, supports pass through messages, supports Publish Data and block mode transfers, and caches several read and write commands. The commands are described in the following subdivisions of 7.5.7.8.

7.5.7.8.2 Required Gateway Commands

The Gateway shall support the Commands that are listed in Table 37. The Publish Data command may not be applicable to the host application interface but, they are required to manage data publishing by the field devices. The same requirement applies to the event notification commands, too.

The I/O system commands shall be supported and employed transparently by the Gateway.

Table 37 – Required Gateway Commands

Device Commands

Cmd	Description
38	Reset Configuration Changed Flag
41	Perform Self Test
42	Perform Device Reset
48	Read Additional Status
59	Write Number Of Response Preambles

Cmd	Description
89	Set Real-Time Clock
94	Read I/O System Client-Side Communication Statistics.
106	Flush Delayed Response Buffers
111	Transfer Service Control
112	Transfer Service

WirelessHART Gateway Commands

Cmd	Description
773	Write Network Id
774	Read Network Id
775	Write Network Tag
776	Read Network Tag
794	Read UTC Time Mapping
814	Read Device List Entries
815	Add Device List Table Entry
816	Delete Device List Table Entry
817	Read Channel Blacklist
818	Write Channel Blacklist
821	Write Network Access Mode
822	Read Network Access Mode
833	Read Neighbor information

Cmd	Description
832	Read Network Information
834	Read Network Topology Information
835	Read Publish Data List
836	Flush Cached Responses for a Device
837	Write Update Notification Bit Mask for a Device
838	Read update notification bit mask for a Device
839	Cancel update notifications
840	Change Notification
841	Read Network Device Identity using Nickname
842	Write Network Device's Scheduling Flags
843	Read Network Device's Scheduling Flags
844	Read Network Constraints
845	Write Network Constraints

7.5.7.8.3 Cached Response Messages

The Gateway shall be able to cache the responses to the Commands listed in Table 38.

Table 38 – Cached Response Messages

	HART Command	Command Descriptor
Cached Response in Gateway upon Read	0	Read Unique Identifier
	11	Read Unique Identifier associated with Tag
	13	Read Tag, Descriptor, Date
	20	Read Long Tag
	48	Read Additional Device Status
	50	Read Dynamic Variable Assignments
Publish Data *	1*	Read Primary Variable
	2*	Read Current & Percent
	3*	Read All Variables
	9*	Read Device Variables and Status (only supported for HART 6 and above)
	33	Read Device Variables
	123	Read Trend – each Publish Data contains all 12 Trend values
	Device specific	Any HART command
Event Notification	119	Read Event Notification Status (Time Stamp + Device Status + Command 48)
Responses Cached in Gateway upon Write confirmation	18	Write Tag, Descriptor, Date
	22	Write Long Tag
	35	Write Primary Variable Range Values
	44	Write Primary Variable Units

7.6 WirelessHART Network Manager

7.6.1 General

Subclause 7.6 specifies the WirelessHART Network Manager. The Network Manager is responsible for the overall management, scheduling, and optimization of the WirelessHART Network. As part of its duties the Network Manager initializes and maintains network communication parameter values. The Network Manager provides mechanisms for devices to join and leave the network. It is also responsible for managing dedicated and shared network resources.

The Network Manager communicates with devices on the WirelessHART Network through the network layer which is described in Clause 6. The commands that the Network Manager uses to setup, monitor, and manage the overall network are specified in Clause 8. The Network Manager is also responsible for collecting and maintaining diagnostics about the overall health of the network. These diagnostics are available to be reported to host-based applications. The diagnostics are also used to adapt the overall network to changing conditions.

The scope of the Network Manager is shown in Figure 57.

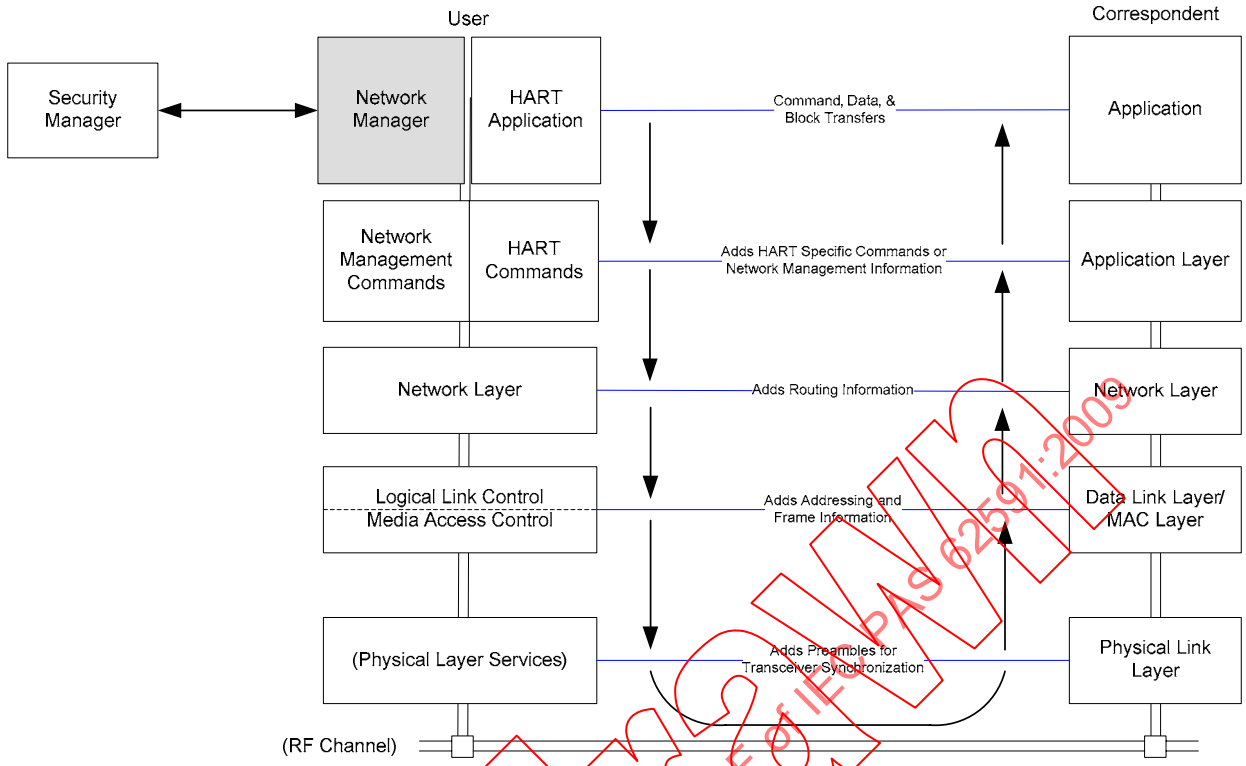


Figure 57 – Network Manager Scope

For the Network Manager to perform its complete set of functions, it needs information about the devices themselves, information about how the network is to be used, and feedback from the network on how well the network is performing. Configuration and setup information about devices is read from the devices themselves. Communication resources are requested by devices, applications, and users. Feedback on how well the network is performing is provided by the devices themselves through health reports and diagnostics. The relationship of the Network Manager to the rest of the WirelessHART Network is illustrated in Figure 58.

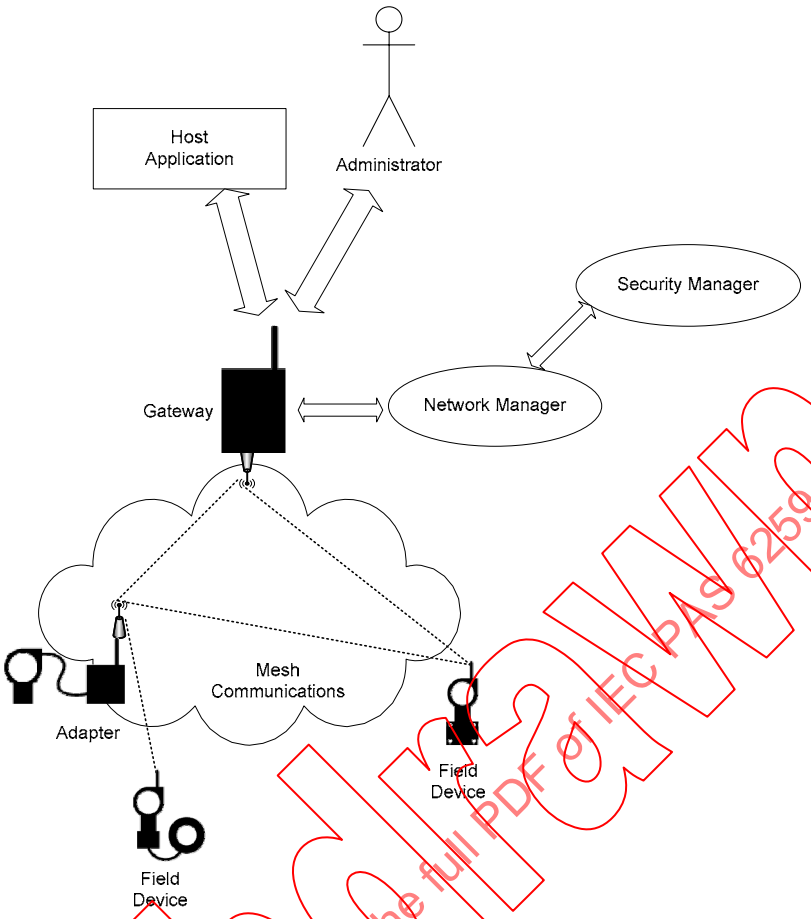


Figure 58 – Network Manager in WirelessHART Network

The user (administrator/maintenance) interacts with the Network Manager Application which generates a network management control packet to Network Devices. Network Management packets travel through the Network layer, then through the Data-Link and the Physical layer before being transmitted through the air to the destination device.

7.6.2 Core Network Functions

7.6.2.1 Network Manager

The Network Manager manages the WirelessHART Network and Network Devices. The Network Manager forms the WirelessHART Network, joins and configures new Network Devices, and monitors the network. The Network Manager uses diagnostic information to adjust the network topology. Since these adjustments are made on an on-going basis the overall operation is referred to as adapting or grooming the network.

The WirelessHART Network architecture does not restrict where the Network Manager resides in the plant automation network. As shown in Figure 48 above, the Network Manager may be co-located with the Gateway in the same box or located in a completely separate physical box. There is one Network Manager per WirelessHART Network.

7.6.2.2 Connection between Network Manager and Security Manager

The Security Manager and the Network Manager are responsible for establishing a connection with each other, and maintaining this connection to support device join requests and establishment of sessions. The connection between the Network Manager and the Security Manager and the method of securing it are not described by the WirelessHART standard. The Security Manager is completely hidden from the Gateways.

7.6.2.3 Security Management

The Security Manager works with the Network Manager to secure the WirelessHART Network from adversarial threats to its operation. The Security Manager generates and manages the cryptographic material used by the network. It is responsible for the generation, storage, and management of keys.

The Security Manager works closely with the Network Manager in a server-client architecture. The Security Manager is shown separately from the Network Manager because it may be a centralized function in some plant automation networks, servicing more than one WirelessHART Network and in some cases other networks and applications. There is one Security Manager associated with each WirelessHART Network. The Security Manager may service multiple WirelessHART Networks.

A secure connection between the Network Manager and Security Manager is required. This secure connection is beyond the scope of this PAS.

7.6.2.4 Network Diagnostics

As part of its system functions, the Network Manager collects network performance and diagnostic information. This information is accessible during run-time, making it possible to view and analyze the behavior of the overall network. If problems are detected, reconfiguration of the network is performed while the network is operating. Network diagnostic information can be accessed through WirelessHART commands.

7.6.2.5 Network Performance

The WirelessHART Network maintains very high reliability through the use of several mechanisms including multiple paths to network devices, multiple RF channels, and multiple communication tries. If improved reliability is required, more paths can be inserted by adding additional network access points and field devices. Additional devices improve path diversity. Additional network access points, and devices in general, increase throughput, reduce latency, and can be used to route around potential interferers.

7.6.2.6 Time-synchronized Communication

All communication on the WirelessHART Network is time-synchronized. The basic unit of measure is a time slot which is a unit of fixed time duration commonly shared by all Network Devices in a network. The duration of a time slot is sufficient to send or receive one packet per channel and an accompanying acknowledgement, including guard-band times for network-wide synchronization. The per-channel qualification indicates that more than one communication can occur in the same time slot.

Precise time synchronization is critical to the operation of networks based on time division multiplexing. Since all communication happens in time slots, the Network Devices shall have the same notion of when each time slot begins and ends, with minimal variation. The WirelessHART protocol defines mechanisms for time synchronization. In the WirelessHART Network, time propagates outward from the Gateway.

7.6.2.7 Sessions

End to end communications are managed on the Network Layer by sessions. Each session contains information on security for a pair (or group) of network devices. All network devices will have two sessions with the Network Manager: one for pairwise communication, and one for network broadcast communication from the Network Manager. All network devices will also have two Network Manager session keys. The sessions are distinguished by the Network Device addresses assigned to them. For the pairwise session with the Network Manager, a device's standard Network Device address will be used; for the broadcast session, a special Network Device address 0xFFFF will be used.

7.6.2.8 Routing

There are two methods of routing packets in a WirelessHART Network: graph routing and source routing.

- Graph Routing: When using graph routing, a Network Device sends packets with a Graph ID in the network layer header along a set of paths to the destination. All Network Devices on the way to the destination shall be pre-configured with graph information that specifies the neighbors to which the packets may be forwarded. In a properly configured network, all devices will have at least two devices in the Graph through which they may send packets. For networks with only one Network Access Point there will also be at least one node with only one outbound path even when properly configured.
- Source Routing: With source routing, pre-configuration of the forwarding devices is not necessary. To send a packet to its destination, the source Network Device includes in the network layer header an ordered list of devices through which the packet shall travel. As the packet is routed, each routing device utilizes the next Network Device address from the packet to determine the next hop to use. Since packets may go to a destination without explicit setup of intermediate devices, source routing requires knowledge of the network topology. Even though no explicit configuration of network devices is required, each hop of the source route requires at least one active link.

7.6.2.9 Connection between Gateway and Network Manager

The interface between a Gateway and the Network Manager is not described by the WirelessHART protocol. The Network Manager and the Gateway are responsible for establishing a secure connection with each other, and maintaining this connection to carry control and data traffic. Thus, it is not necessary for the Gateway to go through the normal network device join process. Once the Gateway connects to the Network Manager, the Network Manager may configure the Gateway to begin advertising to other devices.

There are many forms of Gateways that connect the WirelessHART Network to different physical networks on the plant side. These Gateway Device types include, for example, Ethernet and Serial Gateway Devices. These Gateway Devices are not restricted to any particular protocol.

Ethernet-to-wireless Gateway Device—The Ethernet-to-wireless Gateway Device provides a bidirectional path between industrial Ethernet networks and the WirelessHART Network.

Wi-Fi-to-wireless Gateway Device—A variation of the Ethernet-to-wireless Gateway Device is a Wi-Fi Gateway Device that uses an 802.11a/b/g radio to connect to the plant's network.

Serial-to-wireless Gateway Device—Some plant automation servers and equipment support serial interfaces. A serial-to-wireless Gateway Device connects to serial interfaces of these devices.

Proprietary—Many suppliers have their own IO networks. In these cases, a proprietary-to-wireless Gateway Device connection will be required. Gateways may contain any protocol that suppliers wish to make use of.

A Gateway shall compare the destination address of packets with its own address and the Network Manager's address. Whenever a Gateway receives packets destined for the Network Manager, it may remove the packets from the wireless network and forward them to the Network Manager using its secure connection. Packets with other destinations, as well as packets received from the Network Manager, are routed into the network according to the routing described in the packet.

Once communication paths have been established, the Network Manager is not involved in communications between host applications and network devices. The Gateway is responsible for buffering, protocol conversions, timeouts, time clock, etc.

7.6.2.10 Scheduling

The main functions of the Network Manager are to schedule, monitor, manage, and optimize communication resources. The Network Manager combines information it has about the topology of the network, heuristics about communication requirements, and requests for communication resources from network devices and applications to generate the schedule.

7.6.3 Network Manager Requirements

The Network Manager is central to the overall operation of the WirelessHART Network. The Network Manager is responsible for forming the network, establishing routes, scheduling communication resources, monitoring the health of the network, adapting the network to on-going changes, and working with the Security Manager to allocate and manage Session Keys. The overall set of requirements is summarized below in Table 39.

Table 39 – Network Manager Requirements

Network Function	Requirement
Network Formation and Configuration	Provides logic for initializing itself and starting up the network
	Manages Topology. Understanding the topology of the network. Adapts the network to changes as diagnostic information is reported from devices
	Manages the Network Key. The Network Key is provided to the Network Manager by the Security Manager and is provided to all Network Devices. The Network Manager distributes the Network Key and changes it as required by plant security policies Separate Network Manager and Gateway keys are used for unicast and broadcast traffic that originating from each of them
	Manages Join process. The Network Manager validates devices that wish to join the network. After authenticating a Network Device, the Network Manager gives the joining Network Device the network key and four session keys 1- Network Manager unicast session keys 2 - Network Manager broadcast session key 3 - Gateway unicast session key 4 - Gateway broadcast session key Devices need to be configured with NetworkId independent of the network manager in order to join properly. They need it to find the right network
	Assigns 16-bit Nicknames. The Network Manager assigns and manages network unique 16-bit Nicknames (network addresses) to each Network Device. The Network Manager is responsible for ensuring that the Neighbor Table inside each device is up-to-date
	Establishes a connection with the Gateway. Whenever the Gateway (via the Gateway's Access Points) receive messages destined for the Network Manager, the Gateways forwards them to the Network Manager
	Configures at least one of the Gateway's Access Points to provide the network clock
	Manages network configuration. Maintains a full map of the network configuration, including any information about the network that has been distributed to network devices
	Responds to requests for network information. For example, when a host application makes a request for all of the Network Devices in the network, the Network Manager is responsible for providing the response

Network Function	Requirement	
Routing	Creates and manages network route. The network route is a complete map of the network	
	Manages Neighbor tables. The Network Manager collects network statistics and neighbor table information from each device through periodic health reports. This information is used to adapt the network to changes	
	Builds route tables for Graph routing. Graph Routing is ideal for both scheduled upstream and downstream communications. Upstream communications include process measurements and alarms. Downstream communications include SP changes to actuators.	
	Builds source route lists for Source routing	
	Allocates communication resources to itself, gateways, and to the Devices so that the Network Manager can manage the network and so Devices can communicate	
Network Schedule and Channel Management	Creates Superframes. Multiple Superframes will be used to support communications at specific scan rates. Additional Superframes will be allocated to support device management and diagnostic applications which require large amounts of traffic for short periods of time	
	Assigns links in Superframes	
	Creates Link tables. Each Link includes exactly one slot associated with a Superframe, its type (normal, advertising, discovery) its options (transmit, receive, shared), neighbor information, channel offset, and the device connected to this link	
	Activates and Deactivates Superframes in response to application demands	
	Channel Management	
	Maintains overall WirelessHART Network diagnostic information. For example, when a Network Device has not received a packet from one of its neighbors within the KeepAliveInterval, the device sends a path-down notification to the Network Manager indicating that the path is no longer available	Keeps track of blacklisted channels (blacklisting is a manual operation)
		Provides Channel Offset. The channel offset is used to calculate the channel number when channel hopping. The channel offset takes on a value of 0 to (number of channels less number of blacklisted channels)
Network Diagnostics and Adapting	Maintains record of health information about each Network Device	
	Adapts network to changing environment and application demands. The adaptation includes updating route and schedule information	
	Allocates communication resources as requested by Network Devices. Devices request network capacity to support Publish Data operation, event notification, and block mode traffic. Gateways request bandwidth to support client demands. Network traffic is biased towards a particular path by increasing or decreasing the number of links through a particular device.	
	Optimizes routes and schedules in order to improve operation of the network while conserving power within devices	
	Creation and management of Join Keys	
Security Manager	Creation and management of Session Keys	

The complete Network Management Architecture is shown in Figure 66. The following subclause 7.6.4 describes the components that go into the overall Network Management Architecture. 7.6.4.6 below brings the complete architecture together.

7.6.4 Network Manager Model

7.6.4.1 General Model for Network Manager

7.6.4.1.1 General

The Network Managers' two most important functions are to setup and manage all routes used throughout the WirelessHART Network and to allocate communication resources. The allocation of communication resources is referred to as scheduling. The key components of the Network Manager are the Network Schedule, the collection of Network Devices, the collection of Neighbor Tables, the collection of Connection Tables, and the collection of

Routes. The Network Manager also maintains an association with the Security Manager. The Network Manager is shown below in Figure 59.

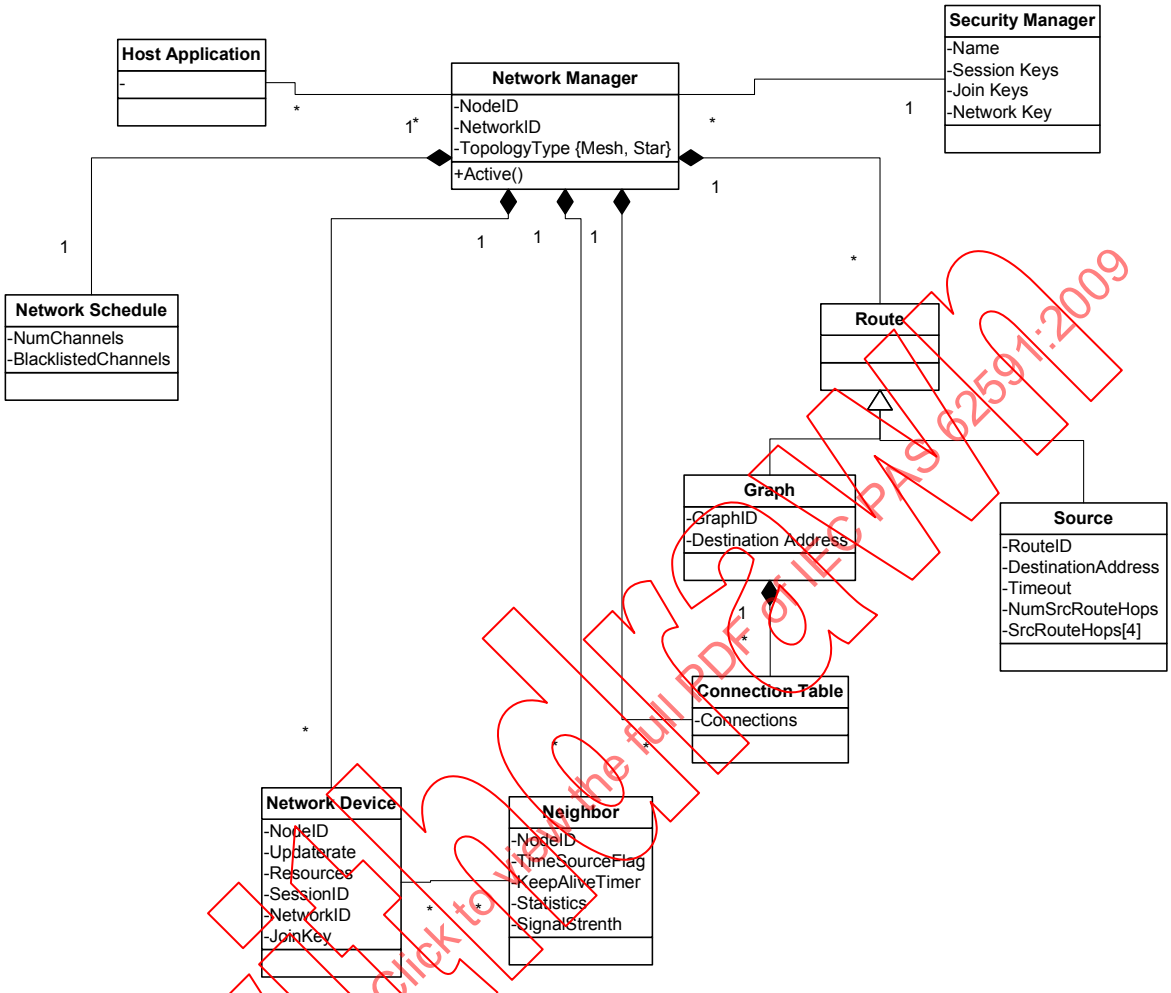


Figure 59 – General Model for Network Manager

In this model, the Network Manager contains one overall Network Schedule. This schedule is further broken down into Superframes and time slots. Time slots are associated with links. The purpose of links is described in detail later in this document (see 7.6.4.4.4). The Network Manager also contains a list of all devices in the network. It also contains the overall network topology including a complete graph of the network and portions of the graph that have been installed into each device. The Network Manager generates route and connection information using information that it receives from the Network Devices. The Graph of the network is built from the list of Network Devices, their reported Neighbors, and current device capacities (e.g., how many descendents the neighbor already has). Each graph should use a maximum of four neighbors as a potential next hop destination.

The Network Manager is also responsible for generating and maintaining all of the route information for the network. The Network Manager uses this route information to generate a complete graph leading from each network device back to the Network Manager. There may also be special purpose routes that are used to send commands and other settings from the Gateways to Network Devices. Finally, there are broadcast routes that are used to send broadcast messages from the Network Manager through the Gateways to all of the Network Devices.

The overall routing information is assembled by the Network Manager using device, neighbor, and diagnostic information reported by the Network Devices. Once the routing information and

communication requirements for each of the devices are known, the scheduling of network resources can be performed. The route the Virtual Gateway is referred to as the "Network Route". Connections are keyed by neighbor ID's.

Special purpose routes are also created. These routes provide paths from the gateways to devices and from devices to gateways – these are referred to as downstream and upstream paths. For example, upstream paths are used for transfer of periodic data from the devices to gateways, and downstream paths are used for transferring setpoint information from the gateway to actuators.

The Network Manager is responsible for adapting the network to changing conditions and for scheduling communication resources. As devices join and leave the network, the Network Manager updates its internal model of the WirelessHART Network and uses this information to generate the schedule and routes. Network performance and diagnostic information is also used by the Network Manager to adapt the overall network to changes in topology and communication requirements. Once the overall schedule has been generated, the schedule is transferred through a series of commands from the Network Manager to the Network Devices.

When devices join the network, the Network Manager is responsible for managing the join process. As part of this join process, the Network Manager uses its model of the network along with algorithms that it has to optimize the network. Once the schedule has been generated, the network and schedule information is transferred through the network in reverse order, i.e., from the Gateways out to each field device.

7.6.4.1.2 Initializing the Network Manager

The Network Manager is responsible for the Network ID – the Network Id shall be configured for each specific WirelessHART Network. One way to initialize the Network ID is through a Gateway that is hosting the Network Manager. In this case the Network Id is configured by connecting the initial Gateway that is hosting the Network Manager to a handheld or host, setting the Network ID, Network Manager Nickname, Network Manager Address, Virtual Gateway Nickname, Virtual Gateway Address, Gateway Join Key, description information, and other parameters.

NOTE The Network Manager and the Virtual Gateway addresses and nicknames are well known. They are specified in Clause 6.

7.6.4.2 Kinds of Network Devices

Network Devices have different behaviors. These behaviors are characterized based on the type of device that they are. Device types and their relationships are shown in Figure 60. These network device types were described earlier in the document (see 7.2.2).

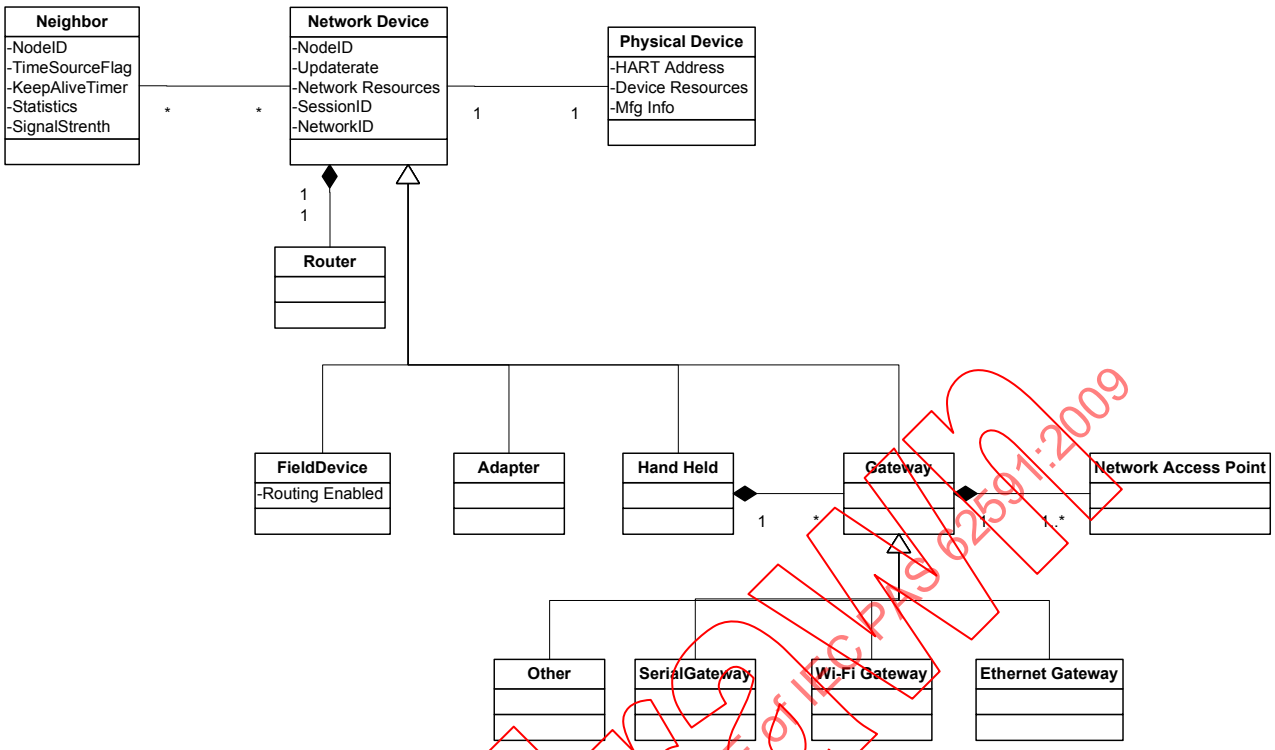


Figure 60 – Kinds of Devices

The root of the device hierarchy is the Network Device. Each Network Device is globally identified by its 5-byte HART Address. The Network Manager contains a complete list of Network Devices. The Network Manager assigns a network unique 16-bit Nickname to each Network Device. Each Network Device also has stored properties containing information on update rates, sessions, and device resources, including items such as the size of the Superframe table, etc. Each Network Device contains a list of Neighbor Devices that it has identified during its listening operations (neighbors can be identified during any open receive time slot – an advertisement packet is special because it contains enough information for a device that needs to join through). There are several more specific device types. Each of these specific device types inherits behavior and properties from the more general Network Device.

The most common type of Network Device is a Field Device. Field Devices can be further differentiated by the type of measurement or control operation they perform – these classifications are not described in this document. All Field Devices shall contain Router capabilities.

Handhelds are used to configure devices, run diagnostics, perform calibrations, and manage network information inside each device. When used in a maintenance lab, Handheld Devices can connect directly to WirelessHART Field devices through their maintenance port.

Routers are used to transfer messages from one location to another. All WirelessHART Field Devices can act as a router.

Gateways are used to connect the WirelessHART Device network to Host applications such as a process automation system or asset management system. The Gateway connects to the WirelessHART Network through Network Access Points.

7.6.4.3 Network Routing

Network Devices use routing to communicate with each other. There are two methods of routing packets in a WirelessHART Network: graph routing and source routing. These are illustrated below in Figure 61.

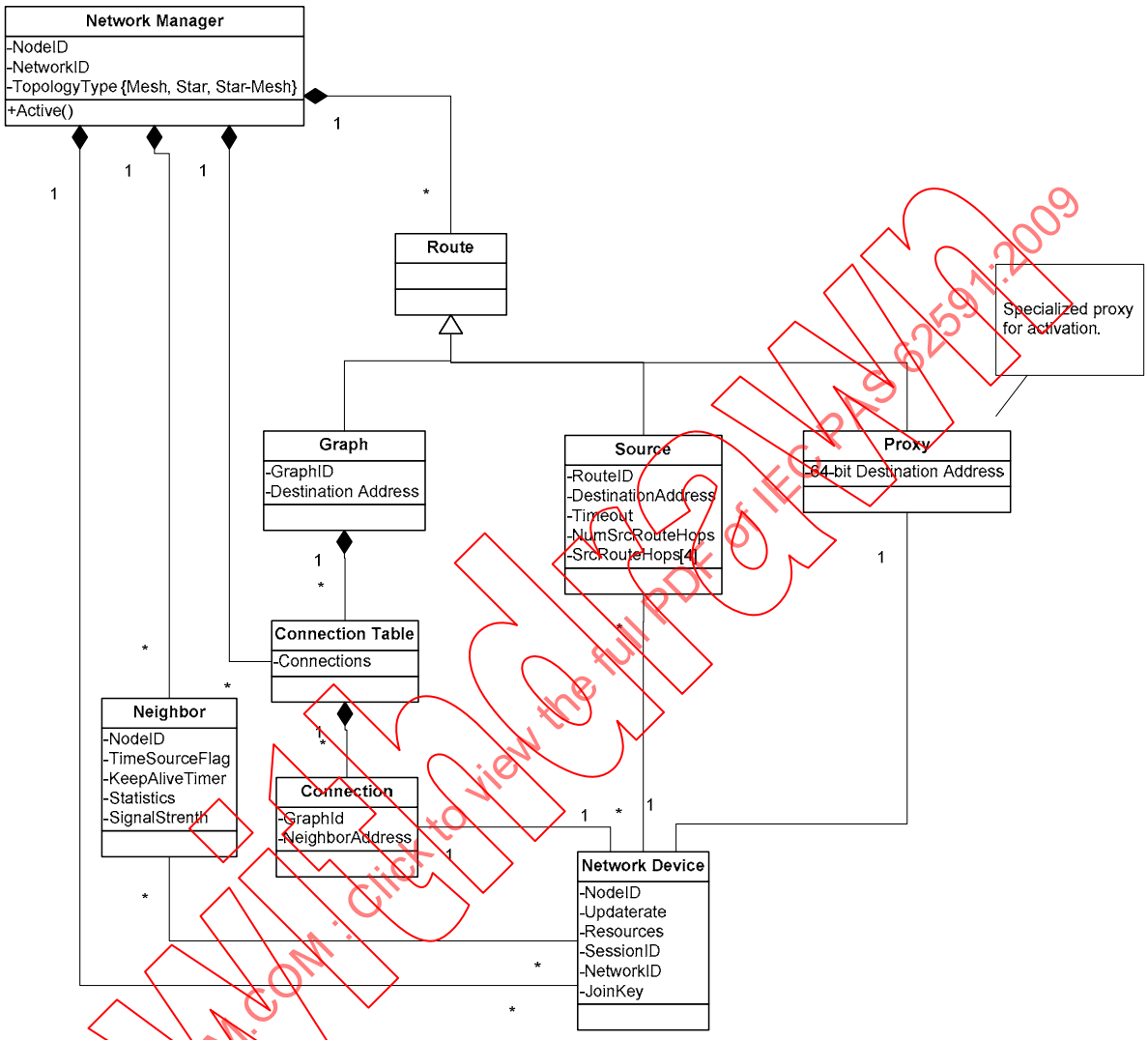


Figure 61 – Network Routing

The Network Manager contains a complete list of Routes, Connections, and Network Devices. When devices are initially added to the network, the Network Manager stores all Neighbor entries including signal strength information as reported from each Network Device. The Network Manager uses this information to build a complete Network Graph. The Network Graph is an optimized route map. During the optization of the Network graph, a large number of possible (but suboptimal) links have been removed. The Network Graph is put together optimizing several properties including reliability, hop count, reporting rates, power usage, and overall traffic flow. A key part of the topology is the list of Connections that connect devices together.

A key function of the Network Manager is to configure Graph and Connection information in each Network Device. The Network Manager maintains a complete list of the Graph and Connection information with which each Network Device has been configured. As the overall network adapts to changing information, the Network Manager is responsible for updating the overall topology, which includes adding and deleting information in each Network Device.

Only the Network Manager knows about source routing. Intermediate devices do not know about Source routes.

Using this knowledge, the Network Manager generates Graph Routes. It also generates Source Routes for the Gateway (Source Routes are best derived from Graph Routes that the Network Manager has already assembled – the graph route is a tree; if you go to each leaf on the tree, then the path from the root to the leaf is a source route). The Network Manager utilizes Graph Routes to generate schedules. This includes measurement information that is transferred from Network Devices to the Gateway and control information that is transferred from Gateway Devices to final control devices such as regulating valves, on-off valves, pumps, fans, dampers, as well as motors used in many other ways.

Every graph in a network is associated with a unique Graph Id. To send a packet on a graph, the source Network Device includes a Graph Id in the packet's network header. The packet travels along the paths corresponding to the Graph Id until it reaches its destination, or is discarded. In order to be able to route packets along a graph, a device needs to be configured with a Connection table. The Connection table contains entries that include the Graph Id and neighbor address. Redundant paths may be setup by having more than one neighbor associated with the same Graph Id. Using Graph Routing, a device routing a packet shall perform a lookup in the connection table by Graph Id, and send the packet to any of the listed neighbors. Once any neighbor acknowledges receipt of the packet (Data-Link level acknowledgement), the routing device may release it and remove the packet from its transmit buffer. If an acknowledge is not received, the device will attempt to retransmit the packet at its next available opportunity.

7.6.4.4 Network Schedule

7.6.4.4.1 General

The Network Manager is responsible for allocating communication resources. The communication resources are divided up into slots and channel offsets and arranged in Superframes. Communication resources are allocated based upon information from the Communication Requirements. The overall model for a Network Schedule is summarized below in Figure 62.

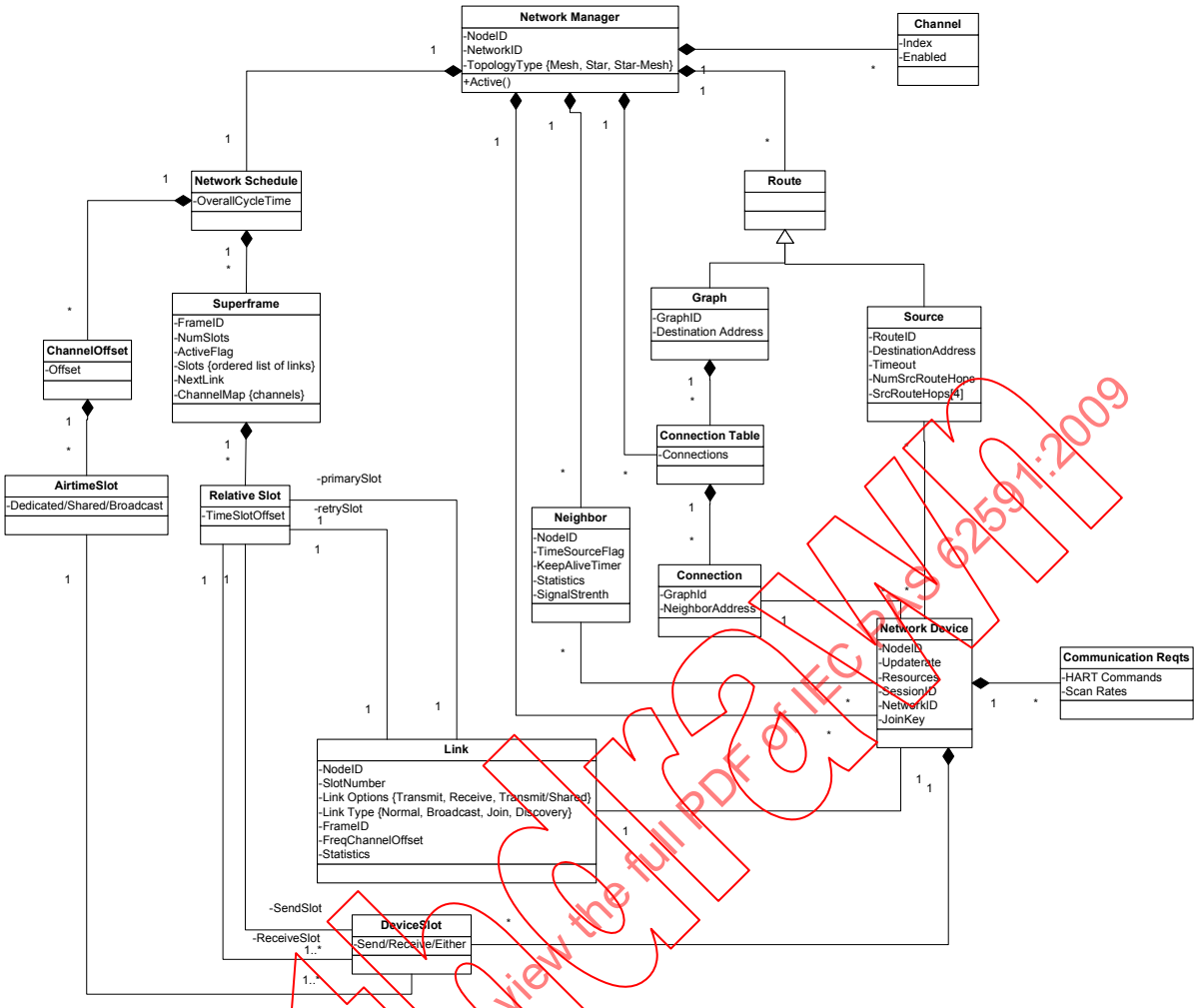


Figure 62 – Network Schedule

Each WirelessHART Network contains exactly one overall schedule that is created and managed by the Network Manager. The schedule is organized into Superframes. Each Superframe is further subdivided into Superframe relative links that repeat as the Superframe cycles. In the drawing above slots in Superframes are called Relative Slots. These slots are relative to the start of the Superframe. Relative Slots shall not be confused with the Absolute Slot Number which indicates the actual time that is being used for transmission of a specific packet.

Link objects are associated with specific Network Device Ids – for each link there are slot allocations in one or more devices. If it is a dedicated slot, then there will be a send slot in one device and a receive slot in another device. If it is a shared slot then there will be a receive slot in one device and one or more transmit slots in several devices. If it is a broadcast slot then there will be one transmit slot in one device and receive slots in several devices. The Link object also contains the Superframe Id, Relative Slot Number, Link Options (transmit, receive, shared), and Link Type (normal, broadcast, advertising, discovery).

The Channel Offset is used to calculate the specific radio frequency channel that is used for a particular slot based on a pseudorandom sequence.

The Network Manager combines the Communication Requirements with the Superframe information to create a set of Links for each device. The specific links loaded into each device are used by the device to determine when the device's radio needs to wake up, and when it wakes up whether it should transmit, receive, or either transmit/receive.

The Link does not determine what is communicated. A Link is an opportunity to communicate. The device determines what it will communicate in each slot.

7.6.4.4.2 Superframe

As noted above, a Superframe is a collection of links assigned to time slots repeating in time. The number of slots in a given Superframe (Superframe size) determines how often each slot repeats, thus setting a communication schedule for devices that use the slots. When a Superframe is created, it is associated with a Graph Id. The Network Manager uses this association to help it allocate Slots and configure Links. In runtime, the device determines how a Link will be used.

Every new Superframe instance in time is called a Superframe cycle. Figure 63 shows how devices may communicate in a simple three slot Superframe. Devices A and B communicate during slot 0, devices B and C communicate during slot 1, and slot 2 is not being used. Every three slots, the link schedule repeats.

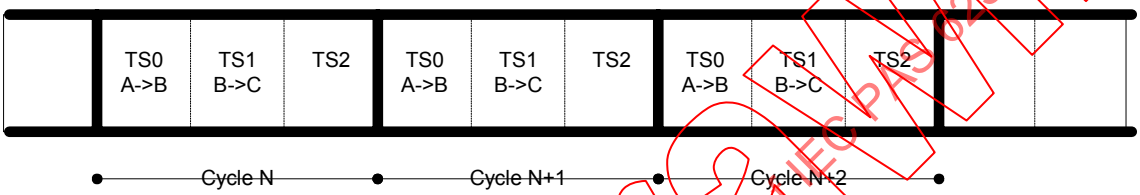


Figure 63 – Example of a Three-slot Superframe

The size of Superframes should follow a harmonic chain, i.e., all periods should divide into each other. Examples of harmonic chains are 1, 2, 4, 8, 16, … and 3, 6, 12, 24 and as well as any other period that conforms to the expression ab^n.

7.6.4.4.3 Multiple Superframes in the Network

A given WirelessHART Network may contain several concurrent Superframes of different sizes. A Superframe is a product of both channels and time slots. Multiple Superframes may be used to define a different communication schedule for various groups of devices or to run the entire network at different duty cycles. Additional Superframes may also be allocated for different communication rates, Publish Data requirements, event notifications, and HART commands issued through host applications.

A Network Device may participate in one or more Superframes simultaneously, but not all devices need to participate in all Superframes. By configuring a Network Device to participate in multiple overlapping Superframes of different sizes, it is possible to establish different communication schedules and connectivity matrices that all work at the same time.

Key applications, such as Asset Management Systems and device specific applications, often require considerable throughput for short durations of time (where short duration is measured in minutes – used to call up configuration and diagnostic screens and respond to user requests). To support this temporary increase in demand for communication time slots, additional Superframes may be used.

Superframes can be added, removed, activated, and deactivated while the network is running (this is important). All Superframes logically start in the same place in time. Cycle 0, slot 0 of every Superframe occurs at the beginning of epoch. The epoch for a specific WirelessHART Network is the time when the Network Manager starts the network. Because of this, time slots in different Superframes are always aligned, even though beginnings and ends of Superframes may not be – this is shown in Figure 64. Because all Superframes begin at the same time, it is always possible to identify time of a given Superframe cycle and time slot.

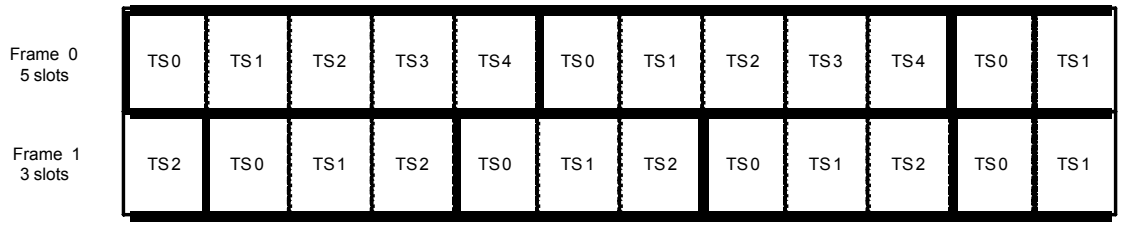


Figure 64 – Multiple Superframes in a Network

A Network Device with links in multiple Superframes may encounter a link arbitration situation. This may happen when two or more Superframes with assigned links coincide in the same absolute time slot. In these cases, the device shall operate on the link that has the numerically lowest FrameId. The rules for link arbitration are defined in the TDMA Data-Link Layer Specification.

7.6.4.4.4 Time Slots

A time slot is a unit of fixed time duration commonly shared by all Network Devices in a network. The duration of a time slot is sufficient to send or receive one packet and an accompanying acknowledgement, including guard-band times for network-wide synchronization.

The model also refers to time slots as 'Relative Time Slots'. This is because time slots in Superframes are always relative to the start of the Superframe.

Dedicated links are shared by a pair of devices that communicate during an allocated time slot. Shared links can have more than one talker and only one listener. For shared links, a defined back-off/ retry mechanism handles collisions that may occur. Broadcast and multicast links have one talker and many listeners, but no Data-Link ACK/NACK acknowledgement. Broadcast links have one talker and some subset of listeners, but no Data-Link level ACK/NACK acknowledgement.

Time slots repeat in time at the rate corresponding to the size of their Superframe. Time slots are assigned to devices through links. If a time slot is assigned to a device, the device can perform one of the following actions within the time slot, depending on the type of link: attempt to transmit a packet, wait to receive a packet, or remain idle. A Network Device that has a transmit link or a transmit/receive link may send a packet during the associated time slot if the destination of the packet matches the neighbor(s) on the other end of the link. A Network Device that has a receive link, or a transmit/receive link with no packet to send, listens for an incoming packet during the associated time slot.

Time slots can also be shared by multiple devices. All devices that participate in either a Dedicated or a Shared Link shall be awake and listening.

7.6.4.4.5 Links

7.6.4.4.5.1 General

When the Network Manager creates connections between devices, link assignments shall be available to support those connections. In many cases it will be necessary for the Network Manager to create new link assignments. In these cases the Network Manger will transfer the link assignments to each of the devices that require it. A link assignment specifies how the Network Device shall use a time slot. Each link includes exactly one time slot, a channel offset, its type (transmit, receive or shared), neighbor information (neighbors are the devices(s) on the other end of the link), and transmit/receive attributes.

7.6.4.4.5.2 Transmit/Receive Attributes

Links may be transmit-only, receive-only, or transmit/receive. A Network Device that has a transmit link or a transmit/receive link may send a packet during the associated time slot if the destination of the packet matches the neighbor(s) on the other end of the link. A Network Device that has a receive link, or a transmit/receive link with no packet to send, listens for an incoming packet during the associated time slot.

7.6.4.4.5.3 Shared Links

Transmit links may be shared by multiple devices, which is indicated to the Network Device by the shared flag in the link configuration. Shared links behave similar to the well-known Slotted Aloha, and devices use a collision-avoidance scheme with a backoff to handle collision situations. Using shared links may be desirable when throughput requirements of devices are low, and/or traffic is irregular or comes in bursts. In some situations, using shared links may decrease latency because the Network Device does not need to wait for dedicated links, but this is true only when chances of collisions are low.

7.6.4.5 Security Manager

The Security Manager is used by the Network Manger to allocate Session Keys. This is shown in Figure 65.

Figure 65 – Security Manager

The Network Manager is responsible for propagating security keys.

7.6.4.6 Detailed Model for Network Manager

Subdivision 7.6.4.6 brings together all the pieces of the Network Manager. The Network Manager is shown in Figure 66 below, along with all of the other key components of the architecture. A memory buffer (PacketQ) and Packet are also included in the diagram. These will be used to help walk through the model from the point of view of routing packets.

The left side of the drawing primarily deals with allocating communication resources. The right side deals with routing. The Network Manager uses routing to determine how to route packets and it uses communication requirements to determine what network resources need to be assigned. Using both of these, the Network Manager generates an overall network schedule, which in effect allocates communication resources in terms of Superframes and Links. Once the schedule has been determined, the Network Manager transfers these configurations items to each Network Device (e.g., by issuing Write Superframe, Write Link, Write Graph).

To illustrate how the WirelessHART network works, consider what happens when a device wakes up (this is simplistic overview, there are many rules not considered here). The device first looks at its list of links and selects the next link. If the Link Option is Receive, the device immediately enters into a listen mode. If the Link Option is Transmit or a Transmit/Receive, the device will enter into transmit logic. Since the Link identifies which device can be transmitted to, the next step is to see if the device has a packet in its PacketQ that can be sent to the device on the other end of the Link. To find a match the device shall look at the routing information in each Packet in the PacketQ. If a match is found the Packet is sent. If a match is not found and the Link Type is Transmit/Receive, then the device will listen for an incoming packet. If no match is found then the device is returned to sleep mode.

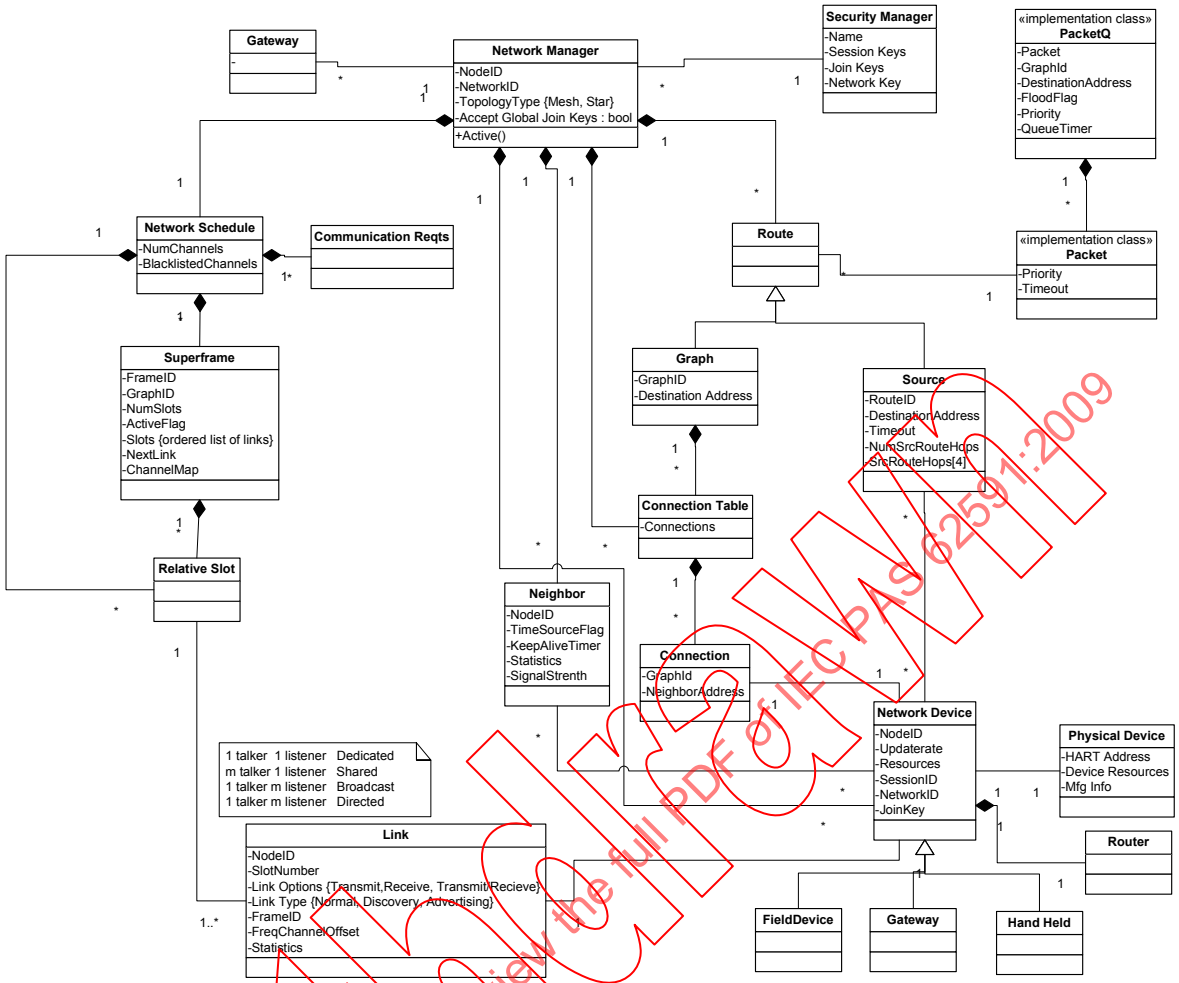


Figure 66 – Network Management Architecture

7.6.4.6.1 Network Addressing

When a Network Device joins the WirelessHART Network, the Network Manager assigns it a 16-bit address (Nickname). The network header of every packet contains source and destination network device addresses, which do not change as packets are routed through the network. It is the responsibility of each network device to terminate packets in which the destination network device address matches the device's own address. Broadcast network device address may be used to address all network devices.

7.6.4.6.2 Network Formation

A key attribute of a WirelessHART Network is its ability to self-organize. There are three components of network formation: advertising, joining, and resource allocation. As part of advertising, Network Devices that are already part of the network send packets announcing the presence of the network that they are part of. Advertisement packets include time synchronization information and a unique NetworkID. Devices trying to join the network listen for these packets and try to match the advertised NetworkID with their own; once at least one advertisement is heard, the new Network Device can attempt to join the network. A new Network Device shall be authenticated to join the network. After a Device has joined the network, it can negotiate with the Network Manager for network resources. The overall Join Sequence is described in Network Management.

7.6.5 Routing

7.6.5.1 General

A key part of the Network Manager's duties is to develop the overall routing for the network. In order to put together efficient and optimized routes, the Network Manager needs information about the network, information about communication requirements, and information about the capabilities of the network devices themselves. As this information is discovered, the Network Manager adjusts the connections in the network until it has a good working system. Subclause 7.6.5 summarizes some of the key requirements, presents rules for determining routes, and outputs a simple example of a route.

7.6.5.2 Routing Requirements

The requirements for the Network Manager are summarized in Table 40.

Table 40 – Routing Requirements

Requirement
Creates and manages network route. The Network Manager maintains an internal representation of the entire network (which in extreme cases could have every node connected to every other node). The Network Manager prunes the route information down into what it believes is a reasonable representation of the network. This internal representation is used to generate Graph and Source Routes
Manages Neighbor tables. The Network Manager collects network statistics and neighbor table information from each device through periodic health reports. The connection and signal level information is used to adjust the routes
Health reports. The communication information is used to choose between existing connections and make decisions on forming new ones
Builds route tables for Graph routing. Graph Routing is ideal for both scheduled upstream and downstream communications. Upstream communications include process measurements and alarms. Downstream communications include SP changes to actuators
Builds source route lists for Source routing
No circular loops in any route (graph or source)
A downstream broadcast graph from the Virtual Gateway to all of the nodes shall be generated
Downstream graphs from the Virtual Gateway to all of the nodes in the network shall be generated

7.6.5.3 Routing Strategy

A basic routing strategy is summarized below. The routing algorithm is not specified in this document.

1) If there is a one hop path to the gateway it should be used.

2) The minimum number of hops to be considered when constructing the graph is 2.

3) The maximum number of hops to be considered when constructing the initial graph is 4.

4) The ratio of the lowest signal strength on a two hop path to the signal strength on a corresponding one hop path for the two hop path should be considered instead of the one hop path.

5) Use the same rule noted in 4) above for 3 and 4 hop paths.

6) The signal level threshold to be used when building the graph; as a first pass 50 % can be used as a starting point. If no paths are found using the specified signal level threshold, then this threshold can be reduced to 0,75 of its previous value and the graph generation retried. This recursion should continue up to four times. If at least one route is still not possible, then it is considered that the node is unreachable.

7.6.6 Scheduling

7.6.6.1 General

The most important thing the Network Manager does is schedule communication resources. In order to put together efficient and optimized schedules, the Network Manager needs information about the network, information about communication requirements, and information about the capabilities of the network devices themselves. As this information is discovered, the Network Manager adjusts the schedule until it has met the requirements. The scheduler then uses feedback from the operation of the system to tune the schedule. Subclause 7.6.6 summarizes some of the key requirements, presents rules for determining routes, and outputs a simple example of a route.

7.6.6.2 Schedule Requirements

The requirements for developing the schedule are summarized in Table 41.

Table 41 – Scheduler Requirements

Function	Requirement	
Assumptions	Network Manager has reasonable representation of network graph	
	Each device has been configured with a connection table	
	Network Manager knows the update rate of each device	
	For redundancy, a datum is configured with one transmit and a retry on one path and another retry on another path	
Constraints	Maximum number of concurrent active channels is determined by the number of enabled channels (limited by black-listing)	
	No devices can be scheduled to listen twice in a slot	
	More than one device can transmit to the same device (e.g. A broadcast link and dedicated links to each of the listening devices can coexist.)	
	On multi-hop path, early hop shall be scheduled first	
	The supported update rates will be defined as 2^n where 'n' is positive or negative integer values e.g., update rate selections of ¼ 250 ms, ½ 500 ms, 1 s, 2 s, 4 s, 8 s, 16 s, 32 s, and 60 s (or more)	
	Base Network Management and Publish Data communications should not exceed 30 % of the available communication bandwidth (100 slots/s max).	
	Services – the network manager shall take into account the service requirements	
	The final schedule (not counting the Gateway spec) should have 50 % free slots (i.e., allocated for retries, listens)	
Data Superframe	The data Superframe length is determined by data scan rate	
	Allocate slots starting with the fastest to the slowest scan rate	
	From the furthest end device, allocate one link for each en-route Network Device to the Gateway. Allocate a 2^{nd} dedicated slot on the same path to handle a retry. Allocate a 3^{rd} shared slot on a separate path to handle another retry	
Management Superframe	Management	Management Superframe has priority over data Superframes
		The network management should be 6 400 slots
		Traverse the graph by breath-first search, starting from the gateway, number the devices as N_0, N_1, … N_n
		At a minimum, every device needs to have a slot for a Keep-Alives and there shall be a corresponding shared receive on the parent side
	Join Process	Join-Request - From the furthest devices, allocate one link for each en-route Network Device to the Gateway (No redundancy provided)
		Join Response - Traverse the graph by breath-first search, allocate one link for each en-route Network Device from the Gateway to end Network Device (No redundancy provided)

Function	Requirement	
		Allocate advertise packets in each device. The number of advertise packets will be inversely related to the number of hops away from the gateway
	Neighbor Discovery	Neighbor discovery. The Network Manager shall allocate discovery link common to all Netowerk Devices. The discoveryInterval timer shall be set to enable discovery
	Network Management Commands	Share the Network Management links with join requests and responses
Command Request/Response Traffic	Allocate shared slots to meet ad-hoc request and response traffic	
Gateway Superframe	The Gateway Superframe should be allocated with a large ID value	
	The Gateway Superframe should be 40 slots long. All slots in the Gateway's Access Points should be allocated)	
	Schedule all unallocated slots in the Gateway. Alternate each of these slots as XMIT, RECEIVE (receive slots shall be shared)	
Special Purpose Superframes	High Throughput	Allocated by gateway or client to address high throughput demand to satisfy asset management and other applications. This will be allocated as a "maintenance" or "block transfer" service type
	Maintenance Superframe	Allocated in the Handheld and Every Field Device. This Superframe is used to provide the Field Device and the Handheld with a high-speed connection to talk on. The Network Manager will allocate 4 slots per s (two links in each direction).

7.6.6.3 Schedule Strategy

A basic scheduling strategy is summarized below. The scheduling algorithm is not specified in this document.

Scheduling Strategy

— Starting from slot 0, the devices are assigned to channel offsets.

— Allocate fastest Publish data requirement first.

— Publish Data destination is always the Virtual Gateway.

Data Superframes

— The data superframe length is determined by data scan rate.

— Allocate slots starting with the fastest to the slowest scan rate.

— From the furthest end device, allocate one link for each en-route Network Device to the Gateway. Allocate a 2nd dedicated slot to handle a rety.

— Each transmission is also scheduled with a retry on another path (if one is available).

— Note one Network Device can only be scheduled to receive once in a slot.

— Event Notification uses same slot allocation scheme as data. If there is Publish Data operation scheduled, then events can share the same slots (they will be sent infrequently – when they are sent they can use the retry slots).

Management Superframe

— Management Superframe has priority over data Superframes.

— The network management Superframe should be (6 400 slots).

— Advertisement

NOTE Advertisement slots are slots that devices use to allow devices wishing to become part of the network to join through.

— Traverse the graph by breath-first search, starting from the gateway, number the devices as N_0, N_1, … N_n

— Every device needs to have a slot for a keep-alive message (If a Network Device has not sent a packet to its parent within this interval, it shall send a KEEP-ALIVE packet).

The keep alive timer is (one slot in the Superframe will be allocated). This should be shared receives on the parent side.

— Each device needs to have three slots every fifteen minutes for health reports.

— Each device needs to have at least one shared slot every minute for request/response requests. If an asset management application is started up, then the Gateway will need to allocate communication resources for that application.

Join Request

— From the furthest devices, allocate one link for each en-route Network Device to the Gateway (No redundancy provided).

Join Response

— Traverse the graph by breath-first search, allocate one link for each en-route Network Device from the Gateway to end Network Device (No redundancy provided).

Network Management Commands and Reponses

— Share the Network Management links with join requests and responses.

Command Request/Response Traffic (e.g., Device Management Request / Response Messages)

— Allocate the links in the same way as join requests.

Maintenance Superframe

— Allocate slots for the Maintenance Superframe – this same Superframe will be set-up in all devices. The Superframe should be 1 s in duration – there are four slots in it.

Gateway Superframe

— The gateway Superframe has a Superframe Id of 253.

— The gateway Superframe is 40 slots long (needs to be a minimum of 400 ms).

— Alternate the slots as XMIT, RECEIVE (should all be Shared).

Optimization

— Number of hops to the gateway

— Alternate paths

— Latency

— Power utilization

— Overall throughput

7.6.6.4 Networking Scheduling Example

In this subdivision a simple scheduling example will be presented. The example covers a WirelessHART Network that consists of one Gateway and three Field Devices. The gateway is identified as 'A' and the three field devices are identified as 'B', 'C', and 'D'. Field Devices B and C communicate every second; Field Device 'D' communicates every 4 s. The arrangement is shown in Figure 67.

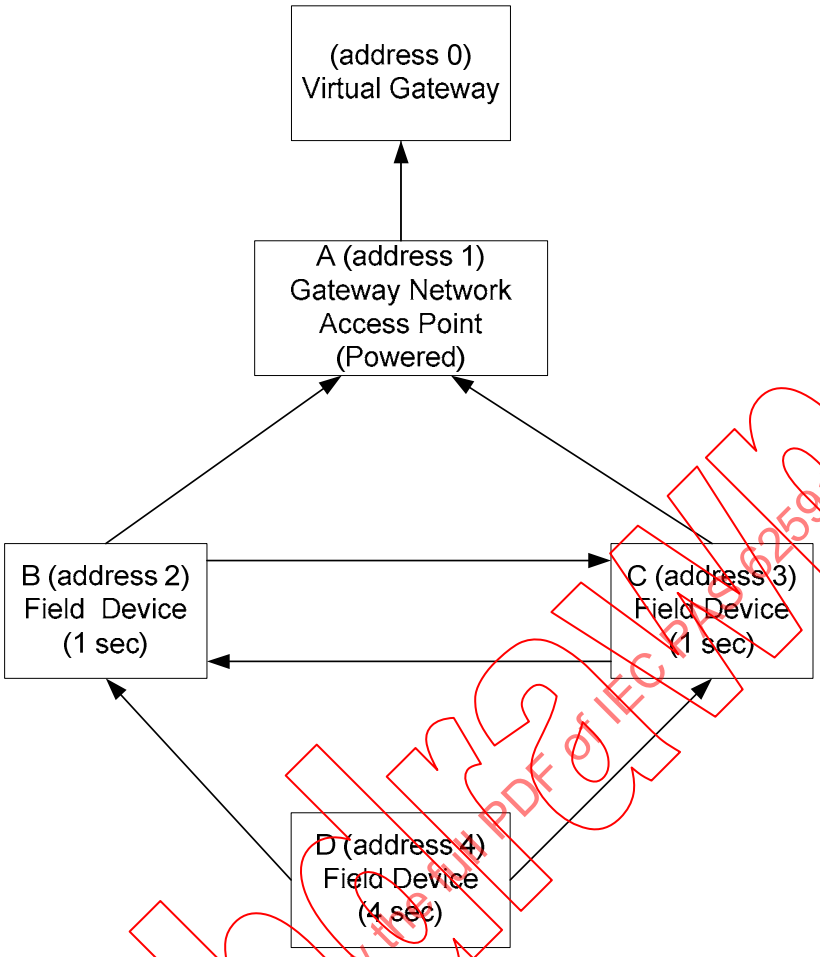


Figure 67 – Example Four Network Device WirelessHART Network

Schedule

In this simple example, the scheduler first creates paths and then allocates communication bandwidth.

Step 1: Select path to the Gateway
B: B->A
C: C->A
D: D->B->A; D->C->A

Step 2: Data Superframes, see Table 42 and Table 43.

Table 42 – FrameId 1: 1 s Update Rate (Superframe Length 100)

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
0	B->A	B->A							
1			C->A	C->A					
2									

Table 43 – FrameId 4: 4 s Update (Superframe Length 400)

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
0			D->B	D->B	D->C				
1					B->A	B->A	C->A	C->A	
2									

Step 3: Management superframe, see Table 44, Table 45, Table 46, Table 47, and Table 48.
FrameId 0: Management Superframe Length 6 000 – once per minute.
Advertisements (advise joining devices about open links to talk on).

Table 44 – FrameId 0: Management Superframe

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
0									*->A
1							*->B		
2	*->C								
3	*->D								
4									

Table 45 – Join Request (shared w/ management responses)

Ch Offset	...	TS7	TS8	TS9	TS10	TS11	TS12	TS13	TS14
0									
1				B->A		B->A			
2					C->A				
3		D->B							
4									

Table 46 – Join Response (shared w/ management requests)

Ch Offset	...	TS11	TS12	TS13	TS14	TS15	...		
0			A->B	A->C	A->B				
1						B->D			
2									

Table 47 – Commands

Ch Offset	...	TS16	TS17	TS18	TS19	TS20	...		
0									
1			B->A	B->A					
2		C->A							
3		D->B							
4									

Table 48 – Command Reponses

Ch Offset	...	TS19	TS20	TS21	TS22	TS23	...		
0		A->B	A->B	A->C					
1				B->D					
2									

Step 4: Create sub-schedules for each node (Link table entries), see Table 49, Table 50, Table 51, and Table 52.

Table 49 – Node A

FrameId	Time Slot	Ch Offset	Device Address	Link Options	Link Type
1	0	0	B	Receive	Normal
1	1	0	B	Receive	Normal
1	2	1	C	Receive	Normal
1	3	1	C	Receive	Normal
4	4	1	B	Receive	Normal
4	5	1	B	Receive	Normal
4	6	1	C	Receive	Normal
4	7	1	C	Receive	Normal
0	8	0	*	Receive	Advertise
0	9	1	B	Receive	Normal
0	10	2	C	Receive	Normal
0	11	1	B	Receive	Normal
0	12	0	B	Transmit	Normal
0	13	0	C	Transmit	Normal
0	14	0	B	Transmit	Normal
0	16	2	C	Receive	Normal
0	17	1	B	Receive	Normal
0	18	1	B	Receive	Normal
0	19	0	B	Transmit	Normal
0	20	0	B	Transmit	Normal
0	21	0	C	Transmit	Normal

Table 50 – Node B

FrameId	Time Slot	Ch Offset	Dest Addr	Link Options	Link Type
1	0	0	A	Transmit	Normal
1	1	0	A	Transmit	Normal
4	2	0	D	Receive	Normal
4	3	0	D	Receive	Normal
4	4	1	A	Transmit	Normal
4	5	1	A	Transmit	Normal
0	6	1	*	Receive	Advertise
0	7	3	D	Receive	Normal
0	9	1	A	Transmit	Normal
0	11	1	A	Transmit	Normal
0	12	0	A	Receive	Normal
0	14	0	A	Receive	Normal
0	15	1	D	Transmit	Normal
0	16	3	D	Receive	Normal
0	17	1	A	Transmit	Normal
0	18	1	A	Transmit	Normal
0	19	0	A	Receive	Normal
0	20	0	A	Receive	Normal
0	21	1	D	Transmit	Normal

Table 51 – Node C

FrameId	Time Slot	Ch Offset	Device Address	Link Options	Link Type
0	0	2	*	Receive	Advertise
1	2	1	A	Transmit	Normal
1	3	1	A	Transmit	Normal
4	4	0	D	Receive	Normal
4	6	1	A	Transmit	Normal
4	7	1	A	Transmit	Normal
0	10	2	A	Transmit	Normal
0	13	0	A	Receive	Normal
0	16	2	A	Transmit	Normal
0	21	0	A	Receive	Normal

Table 52 – Node D

FrameId	Time Slot	Ch Offset	Device Address	Link Options	Link Type
0	0	3	*	Receive	Advertise
4	2	0	B	Transmit	Normal
4	3	0	B	Transmit	Normal
4	4	0	C	Transmit	Normal
0	7	3	B	Transmit	Normal
0	15	1	B	Receive	Normal
0	16	3	B	Transmit	Normal
0	21	1	B	Receive	Normal

7.6.6.5 Process Control Example

Void

Table 53 – Void

7.6.7 Network Manager Interface

7.6.7.1 General

The Network Manager communicates to Network Devices through a series of Wireless Commands. The Network Management uses these commands to read and write communication settings into network devices. For example, the Network Manager can use these commands to create frames and links in devices. In addition the Network Manager shall implement all of the Data Link Layer and the Network Layer Commands.

Each command describes a related set of parameters and has an expected action associated with it. For example, Write/Add/Modify Link contains the parameters needed to add or change a Link. Delete Link is used to remove a link from a Link table in a Network Device. Several commands may be combined into one transaction. For example, the Network Manager could combine commands for creating a superframe and a link.

Write/Add/Modify HART commands add or modify network resources. Delete HART commands are used to terminate use of network items and reclaim resources. Read HART commands are used to read network resources.

The example in Figure 68 shows two message sequences. In the first sequence, the Network Manager sends a Read Superframe command requesting information on a specific Superframe. The Device responds by sending a response. In the second sequence, the Network Manager aggregates and sends commands to modify the Superframe and add Links. The Network Device responds by sending the aggregated results of the commands.

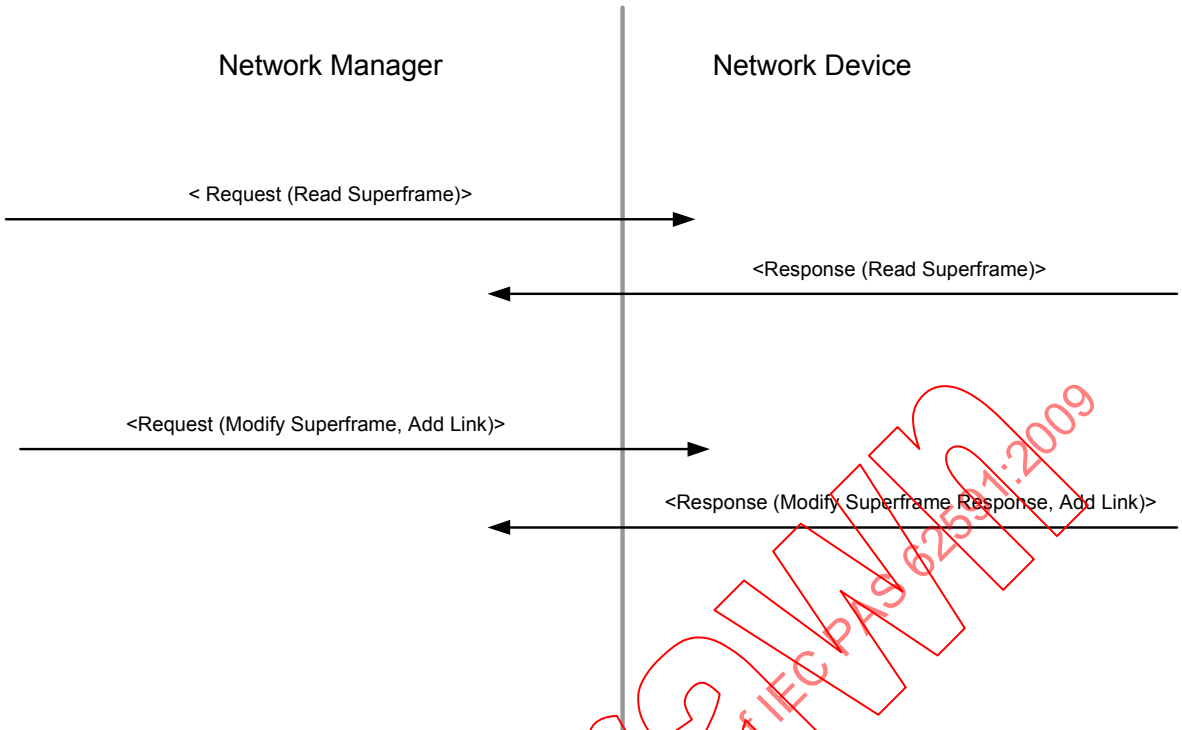


Figure 68 – Example of Command Message Sequences

Scenarios describing how to use the Network Manager interface are summarized in Table 54.

Table 54 – Network Manager Universal Commands

Scenario	Description
Initializing a WirelessHART Network	Starting up a self-organizing WirelessHART Network
Allocating communication resources	Device sends a request to the Network Manager to increase it communication services
Adjusting network schedule	Route is changed and schedule is updated
Health Reports	Device sends a Health Report to the Network Manager
Path failure	A path failure is reported to the Network Manager
Changing a Session Key	Network Manager changes Session Keys
Changing the Network Keys	Network Manager changes Network Keys

7.6.7.2 Initializing a WirelessHART Network

A key characteristic of a WirelessHART Network is its ability to automatically start up and self-organize. Before a WirelessHART Network can form, a Network Manager and a Gateway shall exist and they shall have created a private connection with each other. As part of its initialization sequence the Network Manager will create the following:

1) Network Management Superframe;
2) Network Graph.

Once complete, the Network Manager will activate the first superframe. This establishes the system epoch – ASN 0. The overall initialization sequence is shown in Figure 69.

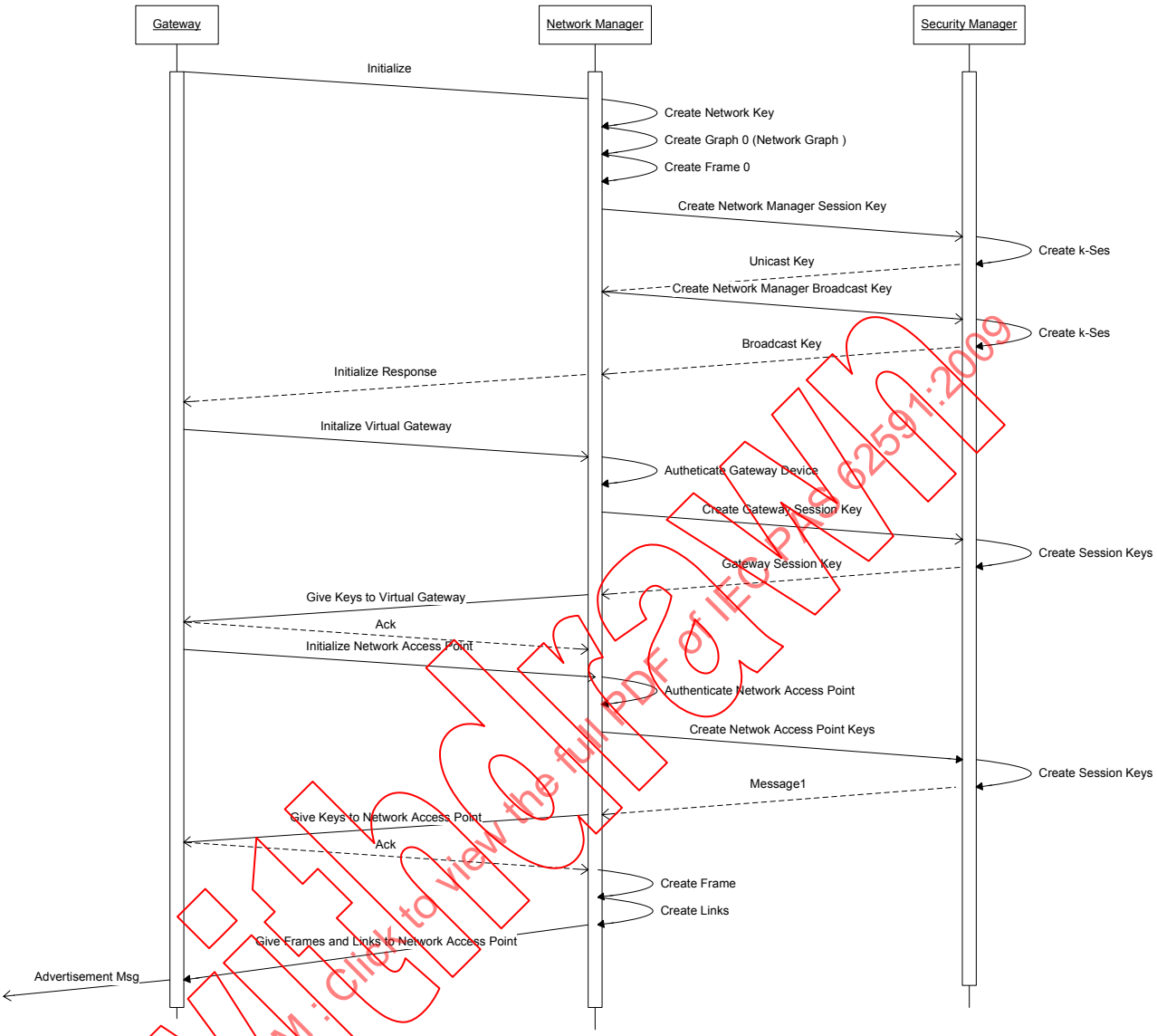


Figure 69 – Initializing a WirelessHART Network

Once the Network Access Point starts to advertise, devices can begin to join the network. As devices join, the network forms. There are three components of network formation: advertising, joining, and resource negotiation. As part of advertising, Network Devices that are already part of the network may send packets announcing the presence of the network. Advertisement packets include time synchronization information and a unique NetworkID. Devices that are trying to join, listen for these packets and try to match the advertised NetworkID with their own. Once at least one Advertisement packet is heard, the new device can attempt to join the network. A new device joins the network by executing a join sequence.

7.6.7.3 Allocating Communication Resources

A device that joins the network shall not start generating or receiving non-network management data until an appropriate amount of network resource is allocated to it. Characteristics of throughput, reliability, and latency associated with a stream of data is called a service. A service may be requested by a device, or may be created by the Network Manager and communicated to the device. A service is identified by service type (Publish, Block Transfer, see Network Management).

To request a service, the device shall send Write/Add/Modify Service to the Network Manager. When the service is created, the Network Manager returns a response code indicating what parameters were granted. If appropriate links/graphs already exist in the network, the Network Manager may do nothing except a response. Otherwise, additional links and/or graphs may be created or activated. This sequence is indicated below in Figure 70.

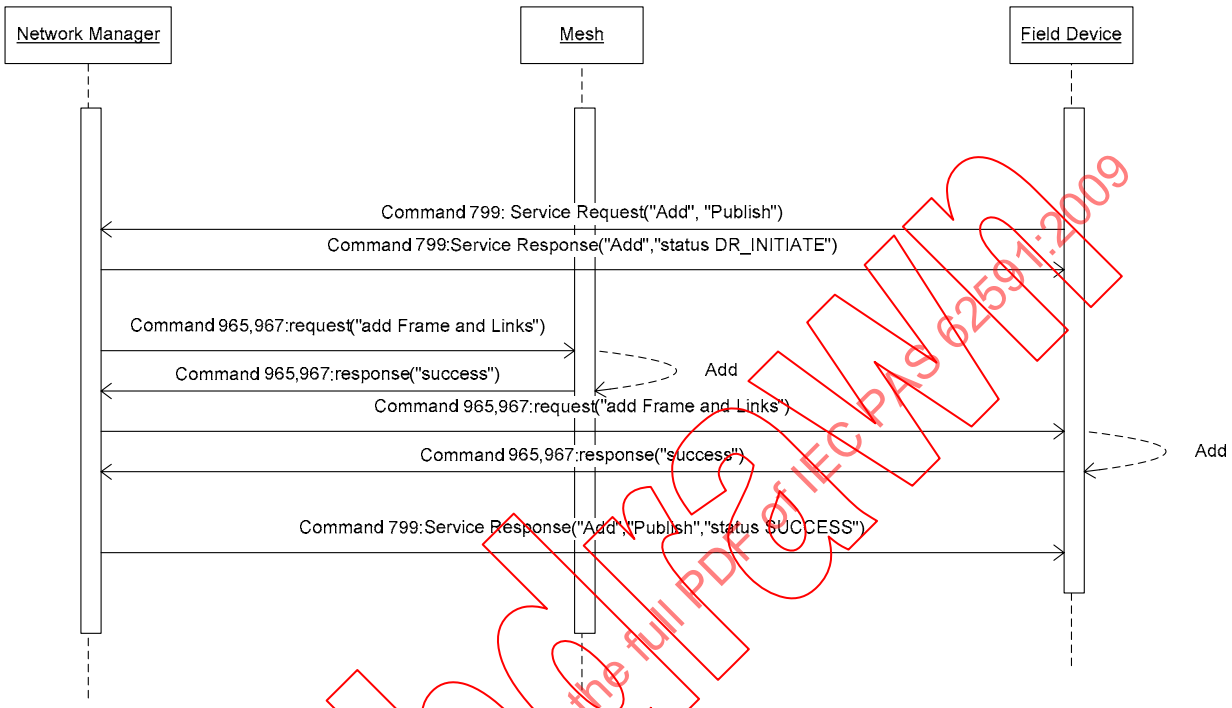


Figure 70 – Allocating and using services

7.6.7.4 Adjusting Network Schedule

The overall network schedule will be adjusted to address changes in routing, device resource usage (e.g., batteries running down), and network demand. This is illustrated below in Figure 71.

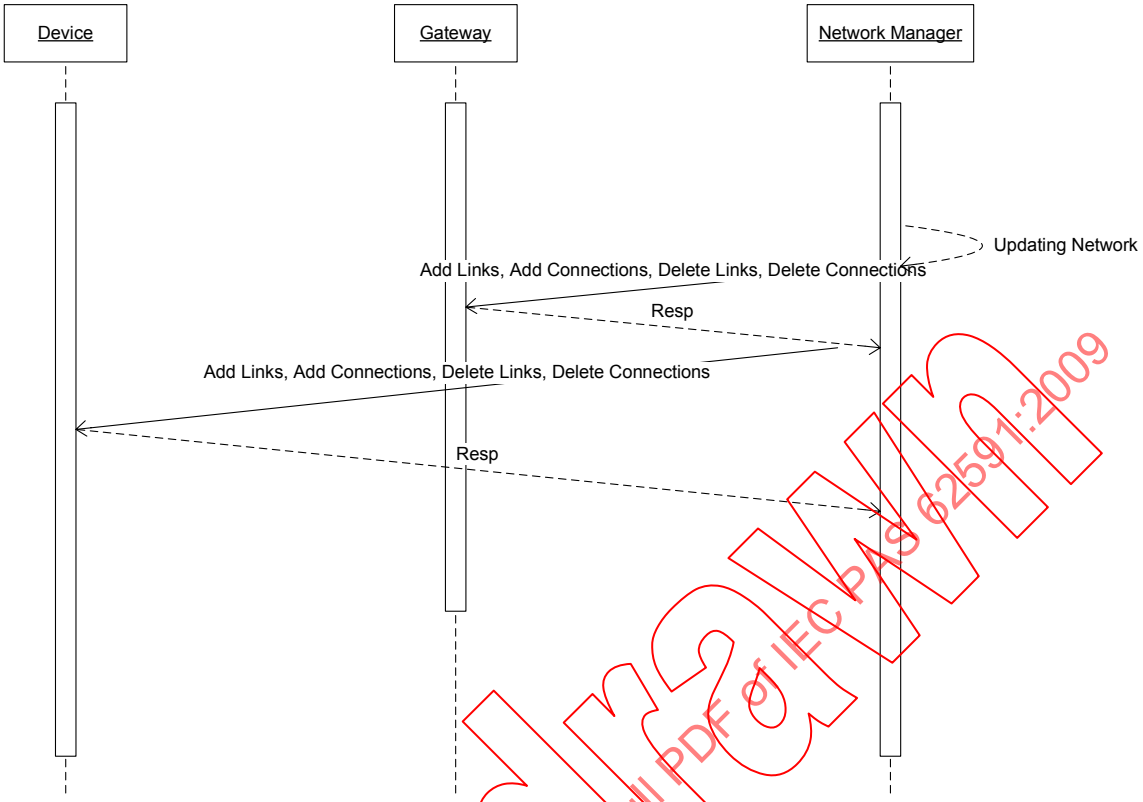


Figure 71 – Adjusting Network Schedule

7.6.7.5 Health Reports

Health reports are transferred from Network Devices to the Network Manager periodically. The Health reports include these messages that are not acknowledged. This is illustrated below in Figure 72. Schedule and network optimization should be performed periodically as well.

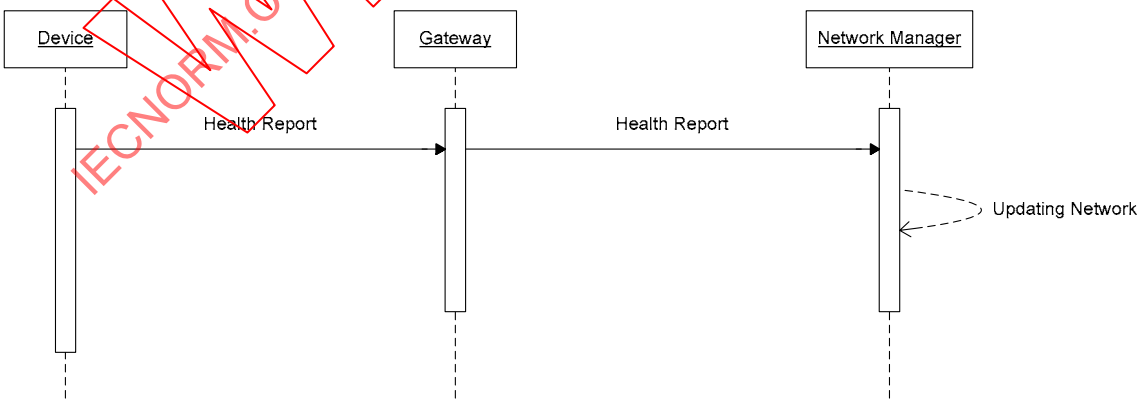


Figure 72 – Health Reports

7.7 Handheld Devices

7.7.1 General

Handheld Devices are used in the installation and maintenance of Network Devices. Handheld Devices are portable equipment operated by the plant personnel. There are four approaches to connect Handheld Devices:

1. **Handheld or application connected through a Plant automation Network**— A plant automation network-connected Handheld Device connects to the plant automation network through some networking technology such as Wi-Fi. This device talks to Network Devices through the Gateway Device in the same fashion as external plant automation servers. To the WirelessHART network this type of handheld is just another host application.

2. **Handheld connected through device maintenance port**—In this mode the Handheld Device connects through the maintenance port interface on the Field Device. When connected in this mode, the Handheld Device cannot talk out through the device into the WirelessHART Network.

3. **WirelessHART Handheld connected to a WirelessHART Network**—A WirelessHART-connected Handheld Device is a device in the WirelessHART Network. In this mode, the WirelessHART Handheld is restricted in the same way as any other device, i.e., it can only talk to the Gateway and to the Network Manager. This mode is used to write keys into the Wireless Handheld and to view diagnostic and system health information. This will be referred to "Connected as a Network Device."

4. **WirelessHART Handheld connected to a WirelessHART Field Device**— A WirelessHART-connected Handheld Device connected over the WirelessHART Network to a WirelessHART Device is restricted to communication with the WirelessHART Device that the handheld is connected to. Special provisioning is used to ensure that the WirelessHART Handheld is restricted to one hop and one device at a time. This will be referred to "Connected as a Maintenance Device."

Out of these scenarios only the last two are of interest in this wireless specification. A WirelessHART Handheld connecting directly to a WirelessHART Device (item 4) is illustrated in Figure 73.

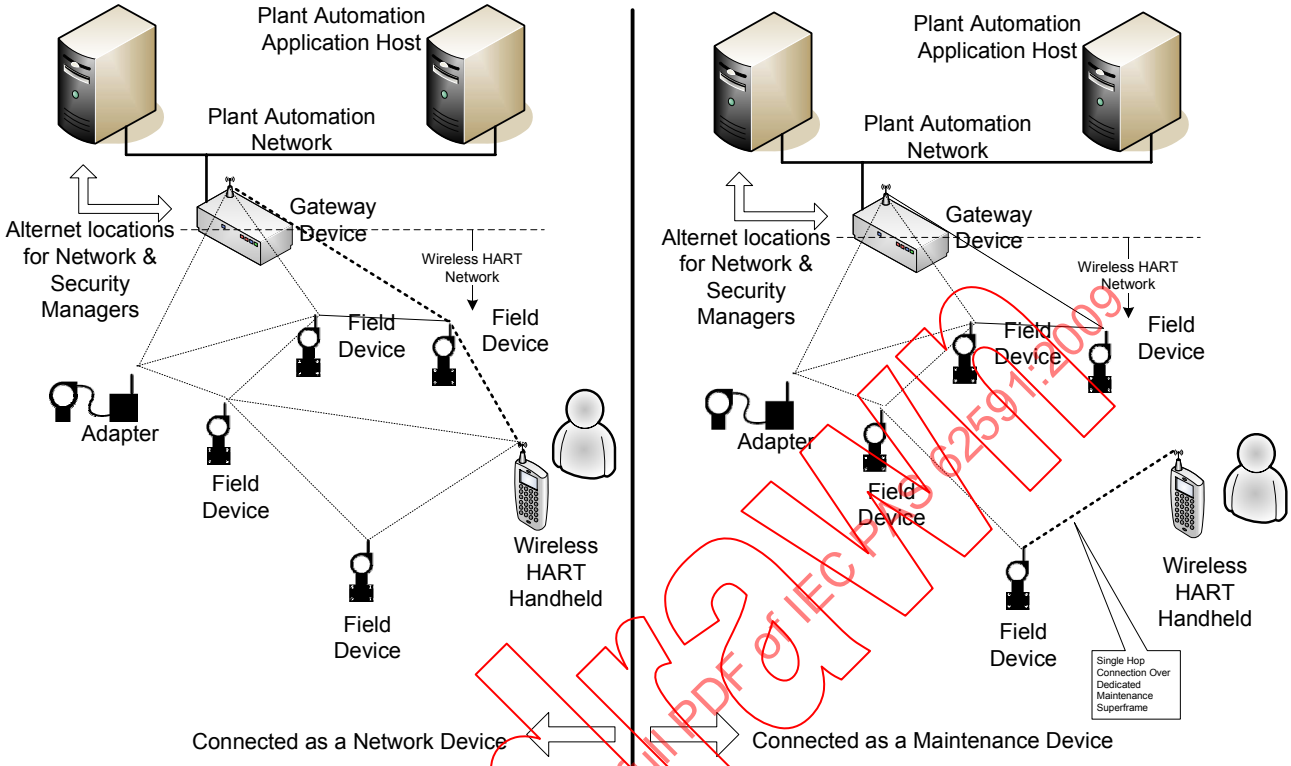


Figure 73 – WirelessHART Handheld Connections

7.7.2 General Requirements

The WirelessHART Handheld shall comply with all WirelessHART requirements.

7.7.3 Maintenance Port Connection

The WirelessHART Handheld can configure and interrogate the device when connected to the maintenance port. All attributes and commands supported by the Field Device are available through the maintenance port and shall be supported by the handheld. When requested, the Field Device answers Identity Commands normally thus allowing the handheld load the EDD for the Field Device. When connected via the maintenance port the handheld does not have access to the wireless network.

7.7.4 Network Device Connection

7.7.4.1 General

There are two reasons to connect a WirelessHART Handheld into the WirelessHART Network as a WirelessHART Device.

1) To install session keys.
2) To view network diagnostics and health reports.

To join a WirelessHART Network as a network device, the user will need to first install the Network Id and the Join Key for the WirelessHART Handheld into the WirelessHART Handheld itself. This can be done in one of several ways including connecting the handheld up to the Gateway via Ethernet to initialize the network and to allocate the initial keys for the WirelessHART Handheld.

Once the handheld has its Network Id and Join key, it will join the WirelessHART Network in the same way as any other device joins the network.

NOTE Initializing the network and the gateway are implementation specific.

7.7.4.2 Install Session Keys

In order for the WirelessHART Handheld to connect to devices in the field, it will need a session key for each device. To get these session keys, the handheld needs to connect to the network as a WirelessHART device and request the Network Manager to allocate session keys for each device it will talk to. The Network Manager will then install these keys into each of the devices and into the handheld. The session keys and their nonce counters will be initialized to the same number at creation.

7.7.4.3 View Network Diagnostics and Health Reports

After the handheld has connected itself to the WirelessHART Network as a network device, it can request the Network Manager to allocate it additional communication resources to talk to the Gateway and it can send requests to the Virtual Gateway and to the Network Manager for diagnostic and Health Report information. The handheld will not be able to talk to devices on the network when connected in this manner (device to device connections are not allowed).

7.7.5 Network Connection as a Maintenance Device

Once a network is up and running, maintenance technicians may want to connect WirelessHART Handheld Devices up to an individual installed Field Device via the WirelessHART Network to gather device specific information, run diagnostics, check device calibration, etc. To support these activities, the WirelessHART Network Manager will allow the WirelessHART Handheld to connect up to one device at a time using a special superframe that has been installed into each device. The sequence for connecting up to the field device is as follows:

1) User connects to the network and obtains sessions (including the keys) for devices it shall talk to.
2) User walks out into plant and gets close to device.
3) Handheld goes into listen mode looking for device advertisement packets.
4) Handheld locates device and identifies links that it can communicate to the device on. It sends the device a command on one of these links requesting additional bandwidth.
5) The device contacts the Network Manager requesting it to activate its Maintenance Superframe.
6) The Network Manager activates the high speed handheld superframe.
7) The device and the handheld now talk on this high speed handheld superframe.

Several things are required to support this scenario.

1) The Network Manager installs a well-known high-speed handheld superframe in each device and into the handheld.

2) The handheld and the device need a peer-peer session key.

3) The Handheld use Commands 806 and 807 to activate and deactivate the Maintenance Superframe.

Once connected to the device, the WirelessHART Handheld device is restricted to talking to the WirelessHART device over the Maintenance Superframe.

7.8 Redundancy

7.8.1 Overview

In many cases, more than one Network Access Point will be used in a WirelessHART Network. When more than one Network Access Point is used, the following shall be considered:

- Routing;
- Synchronization;
- Fault Detection;
- Switch Over.

7.8.2 Routing

The Network Manager is responsible for establishing network routing. Figure 74 shows the relationship of the Network Manager to the WirelessHART Network.

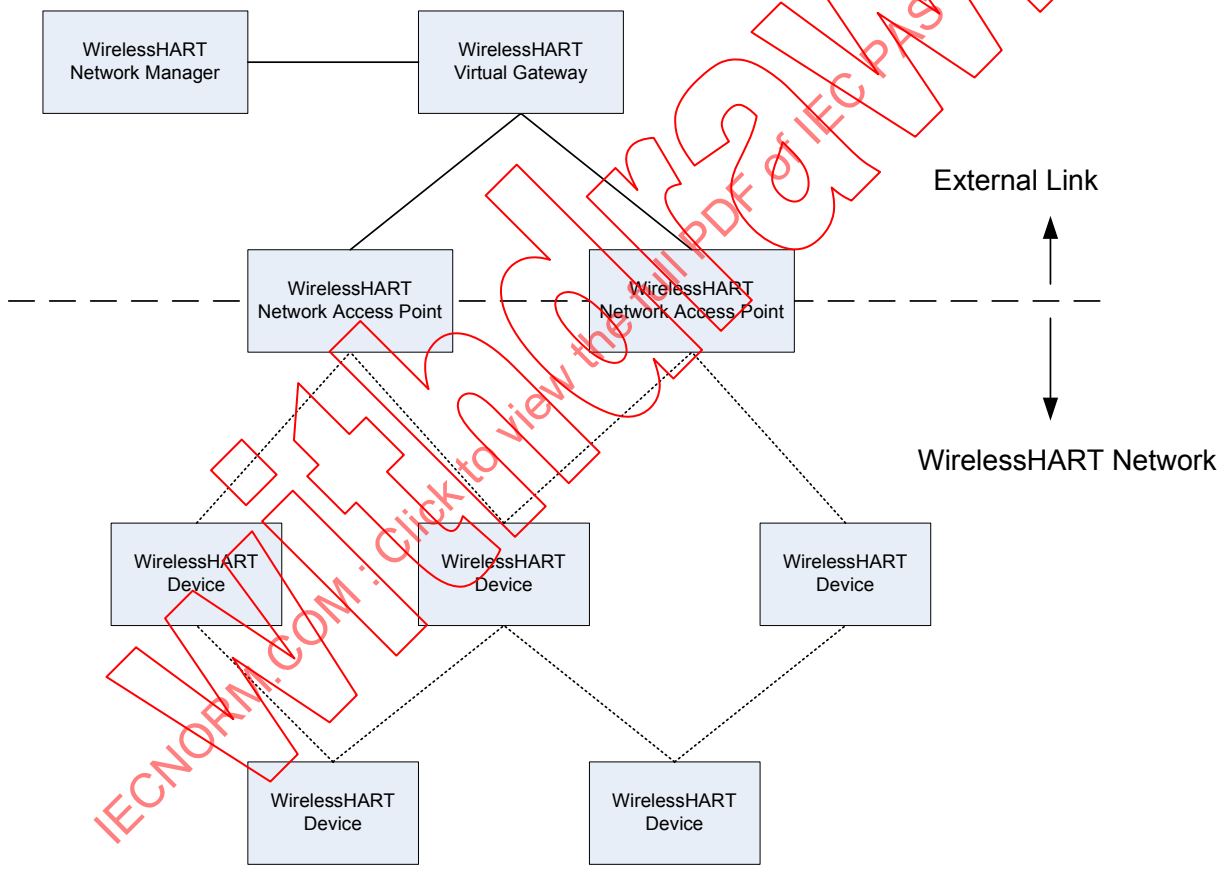


Figure 74 – Network Routing

The Network Manager application will often run in the same physical box or on the same card as the Virtual Gateway. It is also possible to run the Network Manager in a separate Host. In any case, the Virtual Gateway will need to have an external secure communications link to the Network Manager. The WirelessHART specification does not define this connection or the method of securing it.

All WirelessHART devices communicate with the Network Manager as if it were a Network Device with a WirelessHART address.

Every device sending messages to the Network Manager uses a known address, for example Nickname "0", which is discovered during the joining process. Each message traverses the WirelessHART Network to the Virtual Gateway. The message is then routed to the Network Manager. There may be more than one path the message can take to get to the Network Manager. This is illustrated in Figure 75.

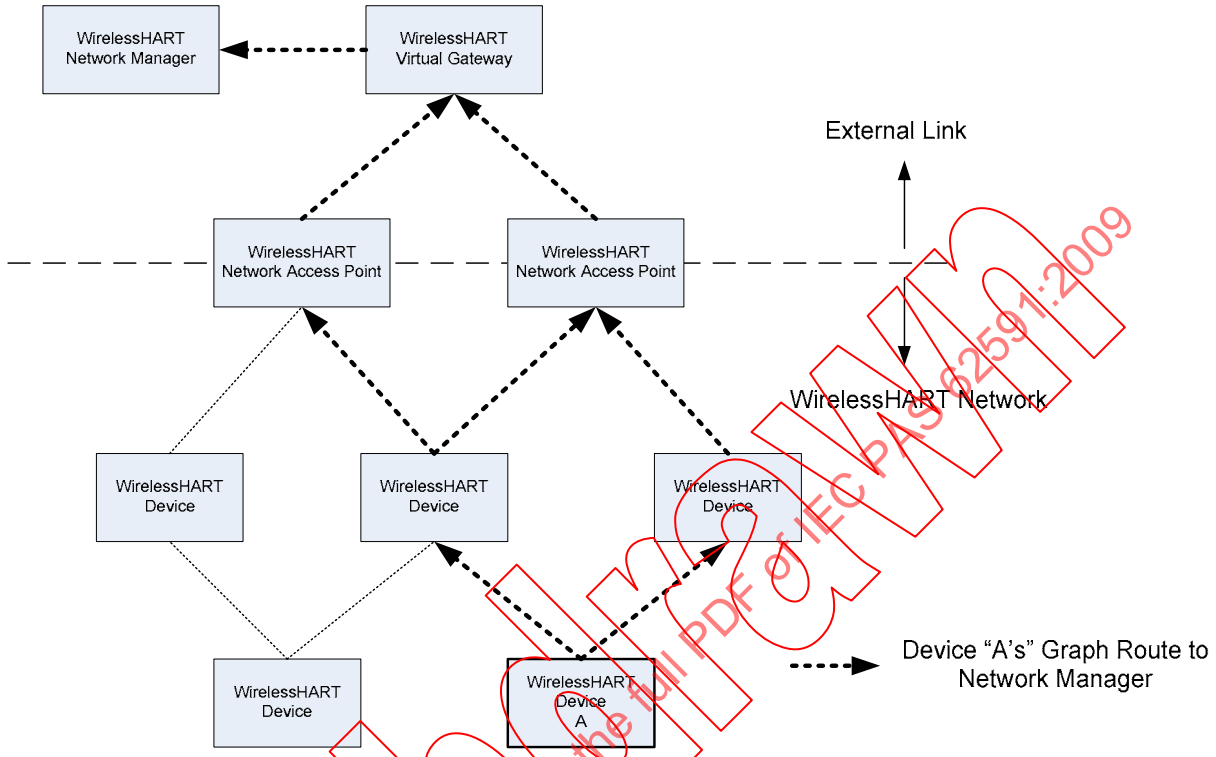


Figure 75 – Graph Routing from WirelessHART Device "A" to the Network Manager

7.8.3 Network Manager Redundancy

Network Manager redundancy can be implemented in a variety of ways. One way to implement Network Manager redundancy is shown in Figure 76. Network Manager Redundancy is outside the scope of the WirelessHART specification.

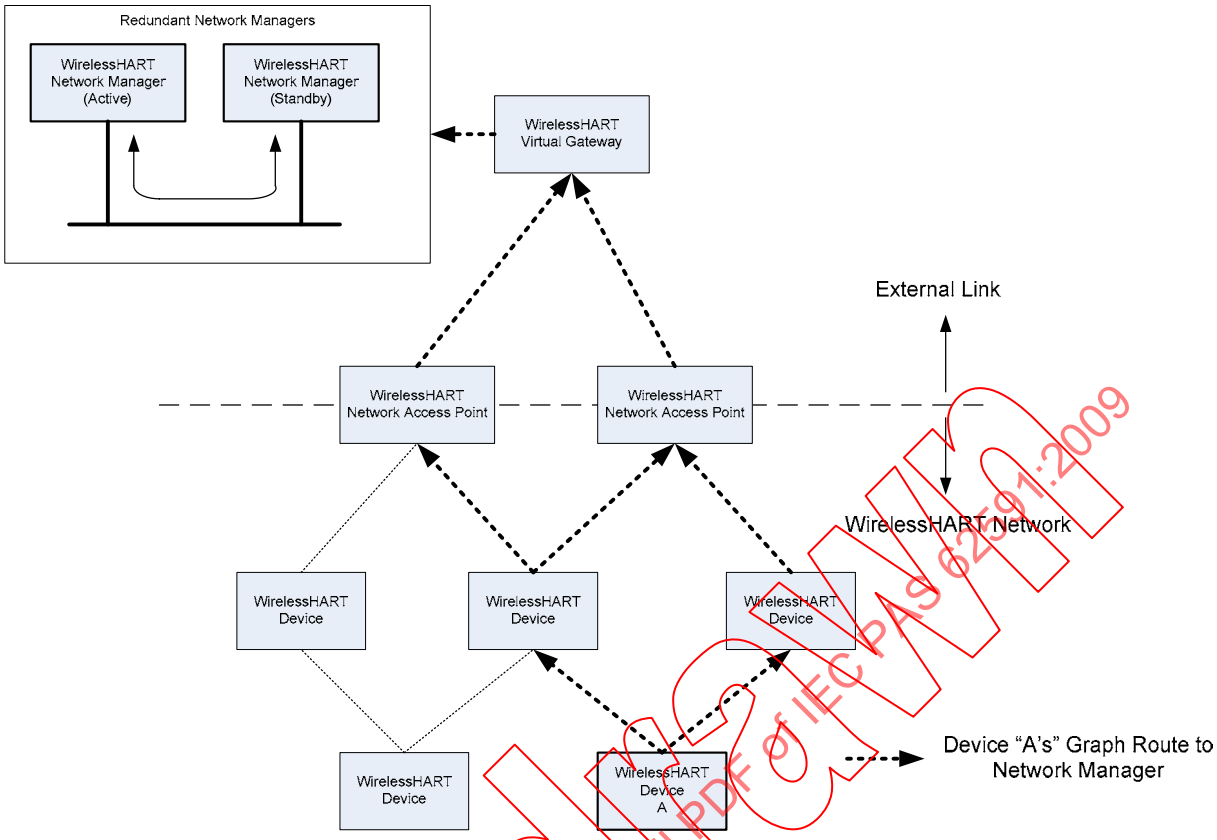


Figure 76 – Redundant Network Managers

The active Network Manager is responsible for keeping the standby Network Manager synchronized. Synchronization can be loose (data bases only) or tight (data bases and state machine states).

There is no WirelessHART Network interoperability issue on how synchronization is done. The one thing we need to be sensitive to is that it is unwise to assume that a synchronization scheme will preserve state in the WirelessHART protocols. For example, if a device is in the process of joining a network. Its state in a partially joined process may be lost on switchover. This means all the WirelessHART processes shall assume that states may be lost and have a recovery mechanism like a time out. This is a wise design philosophy to adopt even if there is no network management switch-over.

Detecting that a redundant Network Management process has failed is key to making the system reliable. Fault detection should be done on both the active and standby units and it may be done through internal and external processes.

Switchover may be initiated by the fault detection or manually initiated.

8 Wireless Network and Gateway Commands

8.1 Overview

This clause on Wireless Commands is a key element of the WirelessHART specifications as it establishes the minimum Application/Network layer support required of all WirelessHART devices.

The Application Layer in HART defines the commands, responses, data types and status reporting supported by the Protocol. While the HCF Enumeration Tables and Response Codes

all establish mandatory Application Layer practices (e.g., data types, common definitions of data items and procedures), the WirelessHART Commands specify the minimum Application/Network Layer content for all WirelessHART compatible devices.

Clause 8 provides application layer command information for WirelessHART Network Manager, Gateway and devices according to Figure 21. Many of these application layer commands are used to control other layers within the protocol stack such as the physical layer, data link layer and the network layer.

8.2 Subject

This specification contains both the definitions and the recommended usage of Wireless Commands. Wireless Commands, if used, shall be implemented exactly as specified. Many Wireless Commands refer to tables from the HCF Enumeration Tables Specification. When HCF Enumeration Tables are referenced, the tables shall be used exactly as specified.

NOTE The HART Communication Foundation Common Tables Specification HCF_SPEC-183 are available at http://www.hartcomm2.org/hcf/services_tools/doc_sales.html.

This specification contains commands that shall be implemented by a Network Manager and Gateway as depicted in Figure 80 hereinafter. It also provides standard commands that shall be implemented for the configuration of each device's Physical Layer, Data Link Layer as well as Network Layer.

8.3 WirelessHART Command Overview

8.3.1 Physical Layer Commands

The commands defined for the physical layer are used by the application layer to control certain features of the wireless physical layer, see Table 55.

Table 55 – Physical layer commands

HART Command	Command Descriptor	Description
797	Write Radio Power	Command to write the radio power output in dBm
798	Read Radio Power	Command to read the radio power output in dBm
804	Read Radio CCA Mode	Command to read the radio Clear Channel Assessment (CCA) mode (Enabled or Disabled)
805	Write Radio CCA Mode	Command to write the radio Clear Channel Assessment (CCA) mode (Enable or Disabled)

8.3.2 Data Link Layer Commands

Implementation of all commands in 8.3.2 is required. These commands control the operation of the Data Link Layer, see Table 56.

Table 56 – DL commands

HART Command	Command Descriptor	Description
773	Write Network Id	Command to write the network id used for the wireless network
774	Read Network Id	Command to read the network id used for the wireless network
781	Read Device Address	Command to read the device addressing information
783	Read Superframe List	Command to read the superframe table entries in a network device
784	Read Link List	Command to read the link table entries in a network device
785	Read Graph List	Command to read the connection of a specific graph
786	Read Neighbor	Command to read specific properties of a particular neighbor

HART Command	Command Descriptor	Description
	Property Flag	
787	Read Neighbor table	Command to read the neighbor table from a network device
788	Alarm Path Down	Command for a device to notify the network manager it had a path issue
795	Write Timer Interval	Command to write various timers required by the data link layer
796	Read Timer Interval	Command to read various timers required by the data link layer
806	Read Handheld Superframe Status	Command to determine if a network device has the maintenance superframe enabled
807	Enable Handheld Superframe	Command for a handheld/maintenance device to request that a device activate the maintenance superframe
810	Read Join Priority	Command to read a network devices current join priority value
811	Write Join Priority	Command to write a network devices' join priority
812	Read Packet Priority	Command to read a network devices current packet priority setting
813	Write Packet Priority	Command to set a network devices packet priority setting
819	Read Back-Off Exponent	
820	Write Back-Off Exponent	

8.3.3 Network Layer Commands

Implementation of all commands in 8.3.3 is required. These commands control the network layer of the WirelessHART device, see Table 57.

Table 57 – Network layer commands

HART Command	Command Descriptor	Description
768	Write Join Key	Command for maintenance on a network device
769	Read Join Status	Command for an application to monitor the Join Process
770	Request Active Advertise	Command for an application to request a network device begin advertising to other devices
771	Force Join	Command for an application to request a device begin the Join process
772	Read Join Mode Configuration	Command for an application to determine a network device's active join mode and shed time
782	Read Session Entries	Command to read the session table of a device
789	Alarm Source Route Failed	Command for the device to notify the network manager it had a source route fail
790	Alarm Graph Route Failed	Command for the network device to notify the network manager it had a failure on a graph route
791	Alarm Transport Layer Failed	Command for the device to notify the network manager that a Transport Layer session failed
799	Request Service	Command for a Network Device to request additional network services for a particular application request
800	Read Service	Command to read the services a network device has allocated
801	Delete Service	Command for a network device to request services no longer required be deleted
802	Read Route	Command for the network manager or an application to read information about a particular route
803	Read Source Route	Command to read the individual hop addresses of a source route
808	Read Time-to-Live	Command to set the packet time-to-live
809	Write Time-to-Live	Command to set the packet time-to-live
814	Read Device List Entries	Read active devices, whitelist or blacklist

HART Command	Command Descriptor	Description
815	Add Device List Table Entry	Adds a device to whitelist or blacklist
816	Delete Device List Table Entry	Deletes a device from whitelist or blacklist
817	Read Channel Blacklist	Command to read the black-list table from a network device
818	Write Channel Blacklist	Command to write an entry into a network devices black-list table

8.3.4 Network Manager Commands

Implementation of all commands in 8.3.4 is required. These commands provide the control for network configuration and management for the wireless network manager and field devices. Network Devices shall only accept these commands from the Network Manager, see Table 58.

Table 58 – Network Manager Commands

HART Command	Command Descriptor	Description
960	Disconnect device	Command for the network manager to remove a device from the network
961	Write Network key	Command for the network manager to write a new network key
962	Write Device Address	Command to write the device short (16-bit) address
963	Write session	Command for the network manager to initiate a session
964	Delete session	Command for the network manager to delete a session
965	Write Superframe	Command for the network manager to write a new superframe entry
966	Delete Superframe	Command for the network manager to delete a superframe entry
967	Write Link	Command for the network manager to write a link table entry
968	Delete Link	Command for the network manager to delete a link table entry
969	Write Graph Connection	Command for the network manager to add a graph connection to a network device
970	Delete Graph Connection	Command for the network manager to delete a previously defined graph in a network device
971	Write Neighbor Properties	Command for the network manager to write the neighbor table of a device and set which neighbor is a clock source
972	Write network suspend	Command for the network manager to suspend transmission of all devices on the network
973	Write Service	Command for the network manager to notify a device of additional service capabilities
974	Write Route	Command for the network manager to write a specific route to a network device
975	Delete Route	Command for the network manager to delete a previously defined route by route id
976	Write Source Route	Command for the network manager to define a source route in a network device

8.3.5 Gateway Commands

Implementation of all commands in 8.3.5 is required for all Gateway Network Devices, see Table 59.

Table 59 – Gateway Commands

HART Command	Command Descriptor	Description
775	Write Network Tag	Command for an application to add a readable tag to identify a specific network
776	Read Network Tag	Command for an application to read the network tag for identification
832	Read Network Device Identity using Unique ID	Command the gateway shall implement to provide the application information about the Network Device
833	Read Neighbor information	Command the gateway shall implement to provide the application the Network Device's neighbor information
834	Read Network Topology Information	Command the gateway shall implement to provide an application with information about the network connections and topology
835	Read Publish Data Mode List	Command returns the Publish Data Mode List that the requested device participates in
836	Flush Cached Responses for a Device	Command instructs the Gateway to flush the cached responses that a device participates in
837	Write Update Notification Bit Mask for a Device	It registers a client for notification updates
838	Read update notification bit mask for a Device	This command asks the Gateway to return the list of update notifications for a Device
839	Change Notification	This request is sent by the Gateway to the Client. The notification lists the changes for the client. Up to 10 change notifications can be included in the response message
840	Read Network Device's Statistics	Returns the number of graphs, frames, and links that a device has currently active
841	Read Network Device Identity using Nickname	Command the gateway shall implement to provide the application information about the Network Device
842	Write Network Device's Scheduling Flags	This command allows users to request special consideration for a device when the Network Manager is creating schedules
843	Read Network Device's Scheduling Flags	Reads the network device properties that a network manager may consider when creating schedules
844	Read Network Constraints	Reads the current setting of the Network Management Strategy
845	Write Network Constraints	Writes the current setting of the Network Management Strategy

8.3.6 Wireless Application Commands

Implementation of all commands in 8.3.6 is required. These are commands that all wireless network devices shall implement, see Table 60.

Table 60 – Wireless Application Commands

HART Command	Command Descriptor	Description
777	Read Wireless Device Information	Command all devices shall implement to provide the application information about the Network Device
778	Read Battery Life	Command all devices shall implement to provide the application the Network Device's battery data
779	Read Device Health Report	Command all devices shall implement to provide the network manager and application with information about the devices communication statistics

HART Command	Command Descriptor	Description
780	Read Neighbor Health Report	Command all devices shall implement to provide the network manager and application information about a devices neighbors
781	Read Device Nickname	Allows the Network Manager and an application to read a Network Device's address
793	Write UTC Time	Command for an application to relate the network time to actual time
794	Read UTC Time	Command to determine a network devices current time

8.4 NETWORK Commands

8.4.1 General

The HART commands in 8.4 are supported by WirelessHART Network Devices.

8.4.2 Command 768 Write Join Key

This command allows a local maintenance tool (e.g., a handheld) to modify a Network Device's join key, see Table 61, Table 62, and Table 63. A device shall return to "Access Restricted" if this command is received via the maintenance port while the device is connected to the network.

Table 61 – Command 768 Request Data Bytes

Byte	Format	Description
0-15	Unsigned-128	Key value

Table 62 – Command 768 Response Data Bytes

Byte	Format	Description
0-15	Unsigned-128	Key value

Table 63 – Command 768-specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1 - 4		Undefined
5	Error	Too Few Data Bytes Received
6	Error	Device-Specific Command Error
7	Error	In Write Protect Mode
8 - 15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-64		Undefined

Code	Class	Description
65	Error	Key change failed
66-127		Undefined

8.4.3 Command 769 Read Join Status

This command allows a host system or handheld device to monitor a field device as it transitions through the WirelessHART joining process. It is intended to assist an instrument technician or service people diagnose a device in the event it has difficulty joining the network. The command is specified in Table 64, Table 65, and Table 66.

Table 64 – Command 769 Request Data Bytes

Byte	Format	Description
None		

Table 65 – Command 769 Response Data Bytes

Byte	Format	Description
0	Enum-4	(Least Significant 4 Bits of Byte 0) Wireless Mode (See HCF Enumeration Table 51), the upper four bits shall be set to zero
1-2	Bits-16	Join Status (See HCF Enumeration Table 52)
3	Unsigned-8	Number of available neighbors
4	Unsigned-8	Number of Advertising Packets Received
5	Unsigned-8	Number of join attempts. Too many join attempts will result in the device considering the join failed
6-9	Time	Join retry timer (indicates the amount of time since the last join request was sent)
10-13	Time	Network search timer (indicates the amount of time listening for first advertisement)

Table 66 – Command 769-specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.4 Command 770 Request Active Advertising

This command allows active, fast advertising to be requested. Generally, this is received locally by a network member from a local maintenance tool (e.g., a handheld) and is propagated to the Network Manager. The Network Manager should configure or turn on fast advertising in the device and one or more of its neighbors. The command is specified in Table 67, Table 68, and Table 69.

Table 67 – Command 770 Request Data Bytes

Byte	Format	Description
0-3	Time	Shed Time

Table 68 – Command 770 Response Data Bytes

Byte	Format	Description
0-3	Time	Shed Time
4-7	Time	Advertising Period
8	Unsigned-8	Number of neighbors advertising in addition to this device

Table 69 – Command 770-specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-2		Undefined
3	Error	Passed Parameter Too Large
4	Error	Passed Parameter Too Small
5	Error	Too Few Data Bytes Received
6	Error	Device specific error
7	Error	In Write Protect Mode
8	Warning	Set to Nearest Possible Value
9-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-64		Undefined
65		Declined (e.g., operator overridden)
66 - 127		Undefined

8.4.5 Command 771 Force Join Mode

This command allows a host system or handheld device to force a field device into active join mode. The device shall stay in the active search mode for at least the Join Shed Time. The command is specified in Table 70, Table 71, and Table 72.

Table 70 – Command 771 Request Data Bytes

Byte	Format	Description
0	Enum-8	Join Mode (see HCF Enumeration Table 61)
1-4	Time	Join Shed Time

Table 71 – Command 771 Response Data Bytes

Byte	Format	Description
0	Enum-8	Join Mode
1-4	Time	Join Shed Time

Table 72 – Command 771-specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-2		Undefined
3	Error	Passed Parameter Too Large
4	Error	Passed Parameter Too Small
5	Error	Too Few Data Bytes Received
6	Error	Device specific error
7		Undefined
8	Warning	Set to Nearest Possible Value
9-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-64		Undefined
65	Error	Force Join Declined
66-127		Undefined

8.4.6 Command 772 Read Join Mode Configuration

Reads the Join Mode and Shed Time. The Join Shed Time is the time a device shall be in active search mode. After this time has expired, the device may go into Deep Sleep/Ultra Low Power mode. The command is specified in:

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Enum-8	Join Mode (see HCF Enumeration Table 61)
1-4	Time	Join Shed Time

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.7 Command 773 Write Network Id

This command configures the device to recognize the proper Network ID. This command can only be issued by the Network Manager or via the maintenance port. All Other sources are responded to with "Access Restricted":

- If the device is not connected to the network, the Network ID is changed immediately. The device should enter power save mode and wait for the Force Join Mode command before beginning to actively search for the network.
- If the device is connected to the Network, then the new ID will be used next Join.

Request Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Network Id

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Network Id

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6	Error	Device-Specific Command Error
7	Error	In Write Protect Mode
8	Warning	Network ID change pending. New Network ID will be used next join
0-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)

Code	Class	Description
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-64		Undefined
65	Error	Invalid Network ID
66-127		Undefined

8.4.8 Command 774 Read Network Id

This is a Wireless Data Link Layer Command.

This command is used to read the current setting of the DLL network id of the device.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Network Id

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.9 Command 775 Write Network Tag

Writes the 32-byte Network Tag. The network tag is a proxy for the Network Id so that the network can be identified in a text form in at the application level.

Request Data Bytes

Byte	Format	Description
0-31	Latin-1	Network Tag

Response Data Bytes

Byte	Format	Description
0-31	Latin-1	Network Tag

NOTE The value returned in the response data bytes reflects the value actually used by the Field Device.

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined

Code	Class	Description
5	Error	Too Few Data bytes received
6	Error	Device-Specific Command Error
7	Error	In Write Protect Mode
8-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-127		Undefined

8.4.10 Command 776 Read Network Tag

Reads the 32-byte Network Tag. The network tag is a proxy for the Network Id so that the network can be identified in a text form in at the application level.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-31	Latin-1	Network Tag

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.11 Command 777 Read Wireless Device Capabilities

This structure is used by the Network Manager to determine operational characteristics of a Network Device. The peak packet rate and recovery time shall be used by network managers to accommodate rechargeable or scavenging devices. Battery devices shall be able to sustain the peak packet rate specified lifetime of the device.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Power Source (See HCF Enumeration Table 44. Device Power Source)

Byte	Format	Description
1-4	Float	Peak packets per second
5-8	Time	Duration at peak packet load before power drained. Shall be at least 1 h. (Set to 24 h if not applicable)
9-12	Time	Time to recover from power drain (Set to zero if not applicable). While recovering, the device shall be able to route 1 packet per second
13	Signed-8	RSL (Receive Signal Level in dB) threshold 'good' connection
14-17	Time	Minimum required Keep-Alive time
18-19	Unsigned-16	Maximum number of neighbors
20-21	Unsigned-16	Maximum number of packet buffers

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1 - 127		Undefined

8.4.12 Command 778 Read Battery Life

This command allows an application or the Network Manager to determine the current state of the battery on a Network Device.

NOTE Careful design analysis of long-term battery performance is required to properly implement this command.

Request Data Bytes

Byte	Format	Description
None		

Request Data Bytes

Byte	Format	Description
1-2	Unsigned-16	Battery Life remaining in days. If the device does not have a battery or other energy storage component then the device may return 0xFFFF.

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.13 Command 779 Report Device Health

This command is periodically published to the Network Manager (see Command 795 and HCF Enumeration Table 43). The command response is transmitted at "Process Data" level priority.

All values returned are counters that are only reset just prior to the device joining the network. The counters are incremented and roll-over to 0 upon overflow.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Number of packets generated by this device since last report
2-3	Unsigned-16	Number of packets terminated by this device since last report
4	Unsigned-8	Number of Data-Link Layer MAC MIC failures detected
5	Unsigned-8	Number of Network Layer (Session) MIC failures detected
6	Enum-8	Power Status (See HCF Enumeration Table 58)
7	Unsigned-8	Number of CRC Errors detected

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1 - 127		Undefined

8.4.14 Command 780 Report Neighbor Health List

This command is periodically published to the network manager (see Command 795 and HCF Enumeration Table 43) providing statistics for linked neighbors. The command response is transmitted at "Process Data" level priority. Command 787 reports discovered neighbors.

The neighbor table is treated as a list with entries being added as neighbors are detected. Neighbors with links to device are at the beginning of the list and neighbors without links at the end.

Packet Statistics returned are counters that are only reset just prior to the device joining the network. The counters are incremented and roll-over to 0 upon overflow.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Neighbor table index
1	Unsigned-8	Number of Neighbor entries read

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Neighbor table index
1	Unsigned-8	Number of Neighbor entries read
2	Unsigned-8	Total number of neighbors
3-4	Unsigned-16	Nickname of neighbor
5	Bit-8	Neighbor Flags (See HCF Enumeration Table 59)
6	Signed-8	Mean RSL (Receive Signal Level in dBm) since last report
7-8	Unsigned-16	Number of Packets transmitted to this neighbor
9-10	Unsigned-16	Number of Packets received from this neighbor
11-12	Unsigned-16	Packets received from this neighbor
13- ...		Number of entries based on response byte 1

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to Nearest Possible Value
9-127		Undefined

8.4.15 Command 781 Read Device Nickname Address

This is a Wireless Network Layer Command.

This command allows the Network Manager and an application to read a Network Device's address.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1 - 127		Undefined

8.4.16 Command 782 Read Session List

This command allows the Network Manager to retrieve information about sessions from a Network Device. Sessions are addressed by their position in the list of sessions on the device, and do not assume a particular implementation. Session indexes may change following addition or deletion of sessions.

"Session Index" or "Number of Entries to Read" may be modified and "Set to Nearest Value" Response Code returned.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Session index
1	Unsigned-8	Number of entries to read

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Session index
1	Unsigned-8	Number of entries read
2	Unsigned-8	Number of active sessions
3	Enum-8	Session type. (See HCF Enumeration Table 48. Session Type Code)
4-5	Unsigned-16	Peer Device Nickname
6-10	Unsigned-40	Peer Device's Unique ID
11-14	Unsigned-32	Peer Device's Nonce Counter Value

Byte	Format	Description
15-18	Unsigned-32	The Device's Nonce Counter Value
19-34		Response bytes 3 - 18 will be repeated up to the number of session entries requested or the number of entries the Network Device has available

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to Nearest Possible Value
9-127		Undefined

8.4.17 Command 783 Read Superframe List

This is a Wireless Command.

This command allows the Network Manager to retrieve information about a Superframe assignment from a Network Device. Superframes are addressed by their position in the list of Superframes on the device, and do not assume a particular implementation. Superframe numbers may change following addition or deletion of Superframes.

"Superframe Index" or "Number of Entries to Read" may be modified and "Set to Nearest Value" Response Code returned.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Superframe index
1	Unsigned-8	Number of entries to read

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Superframe index
1	Unsigned-8	Number of entries read
2	Unsigned-8	Number of active superframes
3	Unsigned-8	Superframe ID
4-5	Unsigned-16	Number of slots in this Superframe
6	Enum-8	Superframe mode flags (See HCF Enumeration Table 47)
7-10 …		Response bytes 3 - 6 will be repeated up to number of entries returned in response byte 1

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to Nearest Possible Value
9-127		Undefined

8.4.18 Command 784 Read Link List

This is a Wireless Command.

This command may be used by the Network Manager to retrieve information about a link entry in a Network Device. Links are addressed by their position in the list of links on the device, and do not assume a particular implementation. Link indexes may change following addition or deletion of links.

"Link Index" or "Number of Entries to Read" may be modified and "Set to Nearest Value" Response Code returned.

Request Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Link index
2	Unsigned-8	Number of Links to read

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Link index
2	Unsigned-8	Number of links read
3-4	Unsigned-16	Number of active links
5	Unsigned-8	Superframe ID
6-7	Unsigned-16	Slot number in the superframe for this link
8	Unsigned-8	channelOffset for this link
9-10	Unsigned-16	Nickname of neighbor for this link (or 0xFFFF if broadcast link)
11	Bits-8	linkOptions (See HCF Enumeration Table 46)
12	Enum-8	linkType (See HCF Enumeration Table 45)
13- ...		Response bytes 5 - 12 will be repeated up to number of entries indicated in response byte 2

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to Nearest Possible Value
9-127		Undefined

8.4.19 Command 785 Read Graph List

This command reads one graph from the Graph Table in the device. The index is for reference only and the referenced graph may change if a graph before it on the list is deleted.

"Graph List Index" may be modified and "Set to Nearest Value" Response Code returned (e.g., if a read pas the end of the list is attempted).

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Graph List index

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Graph List index
1	Unsigned-8	Total number of graphs
2-3	Unsigned-16	Graph ID
4	Unsigned-8	Number of neighbors
5-6	Unsigned-16	Nickname of neighbor
7- ...		Repeat for number of neighbors

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to Nearest Possible Value
9-127		Undefined

8.4.20 Command 786 Read Neighbor Property Flag

This command allows the Network Manager to read the properties of a neighbor on a Network Device.

Request Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname of neighbor

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname of neighbor
2	Bits	Neighbor Flags (see HCF Enumeration Table 59)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-64		Undefined
65	Error	Unknown Nickname
66-127		Undefined

8.4.21 Command 787 Report Neighbor Signal Levels

This command is periodically published to the network manager (see Command 795 and HCF Enumeration Table 43) indicating discovered (but not linked) neighbors. The command response is transmitted at "Process Data" level priority. Command 780 reports linked neighbors.

The neighbor table is treated as a list with entries being added as neighbors are detected. Neighbors with links to device are at the beginning of the list and neighbors without links at the end.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Neighbor table index
1	Unsigned-8	Number of Neighbor entries to read

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Neighbor table index
1	Unsigned-8	Number of Neighbor entries read
2	Unsigned-8	Total number of neighbors
3-4	Unsigned-16	Nickname of neighbor
5	Signed-8	RSL of neighbor in dB
6-8		Repeats (as needed) based on response byte 1

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to Nearest Possible Value
9-127		Undefined

8.4.22 Command 788 Alarm "Path Down"

The command notifies the Network Manager that the path to a neighbor failed. The command response is transmitted at "Alarm" level priority. This command shall be transmitted everytime the pathFailInterval lapses (see Clause 6).

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname of neighbor to which path failure was detected

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.4.23 Command 789 Alarm "Source Route Failed"

This command notifies the Network Manager that a source route failed. The command response is transmitted at "Alarm" level priority every time a source route failure is encountered.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname of unreachable neighbor in the source route
2-5	Unsigned-32	Network-Layer MIC (i.e., the MIC generated using the session key) from the NPDU that failed routing

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.4.24 Command 790 Alarm "Graph Route Failed"

This command notifies the Network Manager that a graph route failed. The command response is transmitted at "Alarm" level priority every time a graph route failure is encountered.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Graph Id of the failed route

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.4.25 Command 791 Alarm "Transport Layer Failed"

This command notifies the Network Manager that a Transport Layer connection failed. The command response is transmitted at "Alarm" level priority every time a session failure is encountered.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname of unreachable peer in the end to ends Transport Layer

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.4.26 Command 793 Write UTC Time Mapping

This is a Gateway Command.

This command allows the network manager to set the mapping of start of ASN 0 to UTC time on a device.

Request Data Bytes

Byte	Format	Description
0-2	Date	HART Date
3-6	Time	Time of Day

Response Data Bytes

Byte	Format	Description
0-2	Date	HART Date
3-6	Time	Time of Day

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.4.27 Command 794 Read UTC Time Mapping

This is a Gateway Command.

This command allows a device (including the Network Manager) to read the mapping of ASN 0 to UTC time.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0-2	Date	HART Date
3-6	Time	Time of Day

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1 - 127		Undefined

8.4.28 Command 795 Write Timer Interval

This is a Wireless Data Link Layer Command.

This command allows the Network Manager to set an interval for a timer on a Network Device.

Request Data Bytes

Byte	Format	Description
0	Enum-8	Timer type. (See HCF Enumeration Table 43. Wireless Timer Code)
1-4	Unsigned-32	Timer interval (in ms)

Response Data Bytes

Byte	Format	Description
0	Enum-8	Timer type. (See HCF Enumeration Table 43. Wireless Timer Code)
1-4	Unsigned-32	Timer interval (in ms)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-2		Undefined
3	Error	Passed Parameter Too Large
4	Error	Passed Parameter Too Small
5	Error	Too Few Data Bytes Received
6	Error	Device-Specific Command Error
7	Error	In Write Protect Mode
8	Warning	Set to Nearest Possible Value
9-15		Undefined
16	Error	Access Restricted (i.e., this command shall be accepted only from the Network Manager)
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-64		Undefined
65	Error	Invalid timer type
66	Error	Invalid timer interval
67-127		Undefined

8.4.29 Command 796 Read Timer Interval

This is a Wireless Data Link Layer Command.

This command allows the Network Manager to read the interval for a timer on a Network Device.

Request Data Bytes

Byte	Format	Description
0	Enum-8	Timer type. (See HCF Enumeration Table 43. Wireless Timer Code)

Response Data Bytes

Byte	Format	Description
0	Enum-8	Timer type. (See HCF Enumeration Table 43. Wireless Timer Code)
1-4	Unsigned-32	Timer interval (in ms)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 64		Undefined
65	Error	Invalid timer type
66-127		Undefined

8.4.30 Command 797 Write Radio Power Output

This is a Wireless Physical Layer Command.

It is used to configure the radio power output for a device or gateway. If the device can not accept the exact value sent from the application, it may respond with "Set to Nearest Possible Value". Devices shall support -10 dBm, 0 dBm and +10 dBm.

Request Data Bytes

Byte	Format	Description
0	Signed-8	Output Power in dBm

Response Data Bytes

Byte	Format	Description
0	Signed-8	Output Power in dBm

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-2		Undefined
3	Error	Passed Parameter Too Large
4	Error	Passed Parameter Too Small
5	Error	Too Few Data Bytes Received
6	Error	Device specific error
7	Error	In Write Protect Mode

Code	Class	Description
8	Error	Set to Nearest Possible Value
9-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-127		Undefined

8.4.31 Command 798 Read Radio Output Power

This is a Wireless Physical Layer Command.

It is used to read the radio power output for a device or gateway.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Signed-8	Output Power in dBm

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.32 Command 799 Request Service

This command is used by wireless device to request connection to another device with specified bandwidth and latency characteristics. Response to this command indicates that the network manager accepted the request and will attempt to process it. Response to the request shall include device-unique service id.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Service ID (supplied by and specific to requesting device)
1	Bits	Service Request Flags (See HCF Enumeration Table 39. Service Request Flags)
2	Enum	Service's Application Domain (See HCF Enumeration Table 40. Service Application Domain)
3-4	Unsigned-16	Nickname of the peer with which the service is requested
5-8	Time	Period (Latency if Intermittent flag set)
9	Unsigned	Route ID for this Service ID

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Service ID
1	Bits	Service Request Flags
2	Enum	Service's Application Domain
3-4	Unsigned-16	Nickname of the peer with which the service is requested
5-8	Time	Period (Latency if Intermittent flag set)
9	Unsigned-16	Route ID for this Service ID

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-3		Undefined
4	Error	Passed Parameter Too Small (period/latency)
5	Error	Too Few Data Bytes Received
6	Error	Device Specific Command Error
7		Undefined
8	Warning	Set to Nearest Possible Value
9-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-64		Undefined
65	Error	Service Request denied
66	Error	Unknown service flag

Code	Class	Description
67	Error	Unknown application domain
68	Error	Unknown nickname
69-127		Undefined

8.4.33 Command 800 Read Service List

This command is used to read details of a service.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Service index
1	Unsigned-8	Number of entries to read

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Service index
1	Unsigned-8	Number of entries read
2	Unsigned-8	Number of active services
3	Unsigned-8	Service ID
4	Bits-8	Service Request Flags (See HCF Enumeration Table 39. Service Request Flags)
5	Enum-8	Service's Application Domain (See HCF Enumeration Table 40. Service Application Domain)
6-7	Unsigned-16	Nickname of the peer with which the service is requested
8-11	Time	Period (Latency if Intermittent flag set)
12	Unsigned-8	Route ID
13- ...		Repeats for the number of entries in response byte 1

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6	Error	Device Specific Command Error
7		Undefined
8	Warning	Set to nearest value (number of entries read)
9 - 127		Undefined

8.4.34 Command 801 Delete Service

This command notifies device of service deletion. The delete operation may be caused by peer's request or because of Network Manager's decision.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Service ID
1	Unsigned-8	Reason. (See HCF Enumeration Table 49. Service Deletion Reason Codes)

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Service ID
1	Unsigned-8	Reason. (See HCF Enumeration Table 49. Service Deletion Reason Codes)
2	Unsigned	Number of Service entries remaining

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data bytes received
6-15		Undefined
16	Error	Access Restricted
17-64		Undefined
65	Error	Entry not found
66	Error	Invalid Reason Code
67	Error	Reason Code rejected, Service not deleted
68-127		Undefined

8.4.35 Command 802 Read Route List

This command allows the network manager to retrieve route used for a service.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Route index
1	Unsigned-8	Number of entries to read

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Route index
1	Unsigned-8	Number of entries read
2	Unsigned-8	Number of active routes
3	Unsigned-8	Number of Routes remaining
4	Unsigned-8	Route ID.
5-6	Unsigned-16	Graph ID
7	Unsigned-8	Source-Route Attached (1=Attached, 0=None)
8-9	Unsigned-16	Number of packets transmitted

Byte	Format	Description
10-13	Time	Time packet last transmitted
14- ...		3-13 repeated for number of entries indicated in response byte 1

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to nearest value (number of entries read)
9 - 127		Undefined

8.4.36 Command 803 Read Source-Route

This is a Wireless Command.

This command allows the Network Manager to read the contents of a particular Source-Route. Broadcast addresses are not legal address values in Source-Routes and shall not be included in any responses for this command.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Route ID

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Route ID
1	Unsigned-8	Number of hops
2-3	Unsigned-16	Nickname hop entry 0
4- ...		Repeated for number of entries indicated in response byte 1

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6- 64		Undefined
65	Error	Entry not found
66-127		Undefined

8.4.37 Command 804 Read CCA Mode

This command allows an application to determine if Clear Channel Assessment (CCA) is enabled on a device.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Enum-8	CCA Mode.0 – disabled, 1 – enabled.

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.38 Command 805 Write CCA Mode

This command allows an application to determine if Clear Channel Assessment (CCA) is enabled on a device.

This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0	Enum-8	CCA Mode. 0 – disabled, 1 – enabled

Response Data Bytes

Byte	Format	Description
0	Enum-8	CCA Mode.0 – disabled, 1 – enabled

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6	Error	Device-Specific Command Error
7	Error	In Write Protect Mode
8-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-127		Undefined

8.4.39 Command 806 Read Handheld Superframe

This is a Wireless Network Layer Command.

This command allows an application to determine if a particular network device has the Handheld Superframe enabled. The Handheld superframe is used between a network device and a Handheld device specifically for maintenance purposes.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Superframe ID
1-2	Unsigned-16	Number of slots in the Superframe
3	Enum-8	Superframe Mode Flags (See HCF Enumeration Table 47)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-8		Undefined
9	Error	No Handheld Superframe
10 - 127		Undefined

8.4.40 Command 807 Request Handheld Superframe Mode

This is a Wireless Network Layer Command.

This command allows a handheld device to request that a Network Device enables the Handheld Superframe so maintenance can be done. The Handheld Superframe is used between a network device and a Handheld device specifically for maintenance purposes.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Superframe ID
1	Enum-8	Superframe Mode Flags (See HCF Enumeration Table 47)

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Superframe ID
1	Enum-8	Superframe Mode Flags

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid Selection (e.g., Invalid superframe mode)
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-15		Undefined
16	Error	Access Restricted

Code	Class	Description
17-127		Undefined

8.4.41 Command 808 Read Packet Time-to-Live

This is a Wireless Network Layer Command.

This command allows an application to determine what the current configuration is for a network device Time-to-Live. The Time-to-Live is a parameter that determines how many hops deep a packet will go before it is discarded.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Currently configured Time-to-Live

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.42 Command 809 Write Packet Time-to-Live

This is a Wireless Network Layer Command.

This command allows the Network Manager to write the packet Time-to-Live. The Time-to-Live determines how many hops deep a packet will go before it is discarded. This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Time-to-Live value

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Time-to-Live value set

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6	Error	Device-Specific Command Error
7	Error	In Write Protect Mode
8	Warning	Set to nearest value

Code	Class	Description
8-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-127		Undefined

8.4.43 Command 810 Read Join Priority

This is a Wireless Network Layer Command.

This command allows an application to determine what the current configuration is for a network device Join Priority. The Join Priority determines what order a network device will be allowed to join. Device with a higher priority will join first.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Join Priority.

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1 - 127		Undefined

8.4.44 Command 811 Write Join Priority

This is a Wireless Network Layer Command.

This command allows the Network Manager write Join Priority. Joining devices to determine which neighbor to send its join request to using the Join Priority. The smaller the number, the better the device is for use when joining.

This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Join priority

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Join priority

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors

Code	Class	Description
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to nearest value
8-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.4.45 Command 812 Read Packet Receive Priority

This is a Wireless Network Layer Command.

This command allows an application to determine what the current configuration is for a network device's receive priority. The receive priority determines what packets a device will respond to and/or forward to other nodes.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Enum-8	Packet Receive priority

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1 - 127		Undefined

8.4.46 Command 813 Write Packet Receive Priority

This is a Wireless Network Layer Command.

This command allows the Network Manager to write receive priority. The receive priority determines what packets a device will respond to and/or forward to other nodes.

This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0	Enum-8	Receive Priority (0 – 3)

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Receive Priority (0 – 3)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received

Code	Class	Description
6-7		Undefined
8	Warning	Set to nearest value
9-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.4.47 Command 814 Read Device List Entries

This command allows an application to retrieve the lists of devices that are indicated in the List ID. The list indices may change due to addition or deletion of entries.

The Gateway and Network Manager shall maintain an Active Device List and support this command. The Network Manager should support a whitelist and a blacklist. All Network Devices should support a blacklist. If a Network Device does not support a blacklist, then it may answer this command with "Command Not Implemented".

Request Data Bytes

Byte	Format	Description
0	Enum	Device List Code (see Command Table 55)
1	Unsigned-8	Number of list entries to read
2-3	Unsigned-16	Starting List index

Response Data Bytes

Byte	Format	Description
0	Enum	Device List Code
1	Unsigned-8	Number of list entries read
2-3	Unsigned-16	Starting List index
4-5	Unsigned-16	Total number of entries in the list
6-11	Unsigned-40	Device Unique ID
…	Unsigned-40	Device Unique ID Repeated up to the number of list entries requested or the number of entries the Network Device has available

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid Selection (i.e. Device List Code not supported)
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to Nearest Possible Value
9-127		Undefined

8.4.48 Command 815 Add Device List Table Entry

This command allows the addition of a device to the indicated list. The Active Device List cannot be modified by this command.

The Gateway and Network Manager shall maintain an Active Device List. Any attempt to add an active device (i.e., on the Active Device List) to the blacklist will generate the response code "Device List Conflict".

This command can be used to move a device between the whitelist and the blacklist. If neither a whitelist nor a blacklist is implemented this command is optional.

Request Data Bytes

Byte	Format	Description
0	Enum	Device List Code (see Command Table 55)
1-5	Unsigned-40	Device Unique ID

Response Data Bytes

Byte	Format	Description
0	Enum	Device List Code (see Command Table 55)
1-5	Unsigned-40	Device Unique ID
6-7	Unsigned-16	Number of List Entries remaining

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid Selection (e.g., Device List Code not supported)
3-4		Undefined
5	Error	Too Few Data Bytes Received
6		Undefined
8	Warning	Device already in list
9-15		Undefined
16	Error	Access Restricted
17-64		Undefined
65	Error	No more entries available
66	Error	Device List Conflict
67-127		Undefined

8.4.49 Command 816 Delete Device List Table Entry

This is a Wireless Command.

This command allows deletion of devices from lists that are maintained in the gateway/network manager. If a device is deleted from the Active Device List, the network manager shall reroute the traffic and disconnect the device.

If neither a whitelist nor a blacklist is implemented this command is optional.

Request Data Bytes

Byte	Format	Description
0	Enum	Device List Code (see Command Table 55)
1-5	Unsigned-40	Device Unique ID

Response Data Bytes

Byte	Format	Description
0	Enum	Device List Code (see Command Table 55)
1-5	Unsigned-40	Device Unique ID
6-7	Unsigned-16	Number of List Entries remaining

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid Selection (e.g., Device List Code not supported)
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-15		Undefined
16	Error	Access Restricted
17-65		Undefined
66	Error	Device List Conflict
67-127		Undefined

8.4.50 Command 817 Read Channel Blacklist

This command reads the current channel blacklist and the blacklist that will be used after the next restart.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Number of bits in current channel map array. This depends on the Physical Layer (e.g., for 2,4 GHz O-QPSK DSSS the array is 16 bits long i.e., 2-bytes)
1-*n*	Bits	Current channel map array. This is an array of bits starting with the least significant bit (bit 0 in 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 channel will be used
n+1–2n	Bits	Pending channel map array. This is an array of bits starting with the least significant bit (bit 0 in 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 channel will be used after the next restart of the virtual gateway/network manager

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-127		Undefined

8.4.51 Command 818 Write Channel Blacklist

This command writes a new channel blacklist to the virtual gateway. This blacklist will come into effect only after the Gateway and the Network Manager are restarted.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Number of bits in new channel map array
1–*n*	Bits	Pending channel map array. This is an array of bits starting with the least significant bit (bit 0 in 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 channel will be used after the next restart of the virtual gateway/network manager

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Number of bits in new channel map array
1–*n*	Bits	Pending channel map array. This is an array of bits starting with the least significant bit (bit 0 in 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 channel will be used after the next restart of the virtual gateway/network manager

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1 – 2		Undefined
3	Error	Passed parameter too large (i.e. number of bits exceeds maximum value)
4	Error	Undefined
5	Error	Too Few Data Bytes Received
6	Error	Device-Specific Command Error
7-15		Undefined
16	Error	Access Restricted
17-64		Undefined
65	Error	Illegal frequency channel bits (e.g. channel 16 for 2,4 GHz 802.15.4 PHY)
66-127		Undefined

8.4.52 Command 819 Read Back-Off Exponent

Reads max back-off exponent (not to exceed 7).

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Maximum Back-Off Exponent

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.53 Command 820 Write Back-Off Exponent

Writes max back-off exponent (not to exceed 7).

This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Maximum Back-Off Exponent (shall be between 4 and 7)

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Maximum Back-Off Exponent

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-2		Undefined
3	Error	Passed Parameter Too Large
4	Error	Passed Parameter Too Small
5	Error	Too Few Data Bytes Received
6	Error	Device specific error
7	Error	In Write Protect Mode
8	Error	Set to Nearest Possible Value
9-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-127		Undefined

8.4.54 Command 821 Write Network Access Mode

This command writes the network access mode. This can be used to restrict the access to the network. Only the Network Access Codes 0 and 4 are mandatory.

NOTE The access mode does not substitute the check for the device's credentials such as join key, tag or device id. The access mode provides additional checks.

Request Data Bytes

Byte	Format	Description
0	Enum	Network Access Mode Code (see HCF Enumeration Table 56)

Response Data Bytes

Byte	Format	Description
0	Enum	Network Access Mode Code

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid Selection
3-4		Undefined
5	Error	Too Few Data Bytes Received
6	Error	Device-Specific Command Error
7	Error	In Write Protect Mode
8-15		Undefined
16	Error	Access Restricted
17-31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33	Error	DR Initiated
34	Error	DR Running
35	Error	DR Dead
36	Error	DR Conflict
37-127		Undefined

8.4.55 Command 822 Read Network Access Mode

This command reads the network access mode.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Enum	Network Access Mode Code(see HCF Enumeration Table 56)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-5		Undefined
6	Error	Device Specific Command Error
7–31		Undefined
32	Error	Busy (A DR Could Not Be Started)
33 - 127		Undefined

8.4.56 Command 823 Request Session

This is a command from a Handheld to the Network Manager.

This command requests a session from a handheld to a device. The session is unicast. Only authorized devices shall be granted a session.

Request Data Bytes

Byte	Format	Description
32-33	Unsigned-16	Peer Device Nickname

Response Data Bytes

Byte	Format	Description
32-33	Unsigned-16	Peer Device Nickname
34-37	Unsigned-32	Peer Device's Nonce Counter Value
38-53	Unsigned-128	Key value

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too few data bytes received
6-15		Undefined
16	Error	Access restricted
17-64		Undefined
65	Error	Unknown nickname
66	Error	Field device has insufficient capacity to support another session.
67-127		Undefined

8.5 Gateway and Network Manager Commands

8.5.1 Command 832 Read Network Device Identity using Unique ID

This is a Gateway Command.

This command returns the identity information for the indicated device

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of the device

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of the device
5-6	Unsigned-16	Nickname
7-38	Latin-1	Long Tag

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid selection
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-127		Undefined

8.5.2 Command 833 Read Network Device's Neighbor Health

This is a Gateway Command.

It returns information regarding the quality of the connection to each active neighbor or potential neighbor a device has. By reading Gateway Command 814, the application layer can determine how many neighbors a particular device has in its neighbor list. This command can then be issued as many times as needed to determine the specific information about each neighbor connection.

If an application makes requests for this information from each device attached to a Wireless Gateway, then a picture/graph of the quality of the network can be built.

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of Network Device
5	Unsigned-8	Neighbor index number
9	Unsigned-8	Number of neighbor entries to read

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of Network Device
5	Unsigned-8	Neighbor index number
9	Unsigned-8	Number of neighbor entries returned
10-11	Unsigned-16	Neighbor 1 nickname (2 byte address)
12	Signed-8	RSL of communication received at this device from Neighbor 1
13-16	Unsigned-32	Packets transmitted to neighbor 1
17-20	Unsigned-32	Failed transmits to neighbor 1 - number of packets expecting an ACK and none was received
21-24	Unsigned-32	Packets received from neighbor 1
25-…		Repeat bytes 10–24 for each neighbor returned

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid selection
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-7		Undefined
8	Warning	Set to nearest value
9-64		Undefined
65	Error	Invalid Neighbor table index
66-127		Undefined

8.5.3 Command 834 Read Network Topology Information

This is a Gateway Command. It returns the Graph Id's that the requested device participates in. By reading Command 814, the application layer can determine how many graphs a particular device is participating in. This command can then be issued as many times as needed to determine the specific graph id's that the device is participating in.

If an application makes requests for this information from each device attached to a Wireless Gateway, then a picture/graph of all the network topology can be built.

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Long address of the device information is being requested
5-6	Unsigned-16	Graph index number

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Long address of the device information is being requested
5-6	Unsigned-16	Graph index number
7-8	Unsigned-16	Total number of Graphs in this device
9-10	Unsigned-16	Graph Id for this index
11-12	Unsigned-16	Number of Neighbors returned
13-14	Unsigned-16	Neighbor 1
…		Neighbor *n* (based on number of neighbors id's returned)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid selection
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-64		Undefined
65	Error	Entry not Found
66-127		Undefined

8.5.4 Command 835 Read Publish Data Message List

This is a Gateway Command.

It returns the Publish data Mode List that the requested device participates in. By reading Gateway Command 835 the application layer can determine how many publish data mode commands a particular device is participating in.

If an application makes requests for this information from each device attached to a Wireless Gateway, then a list of all Publish data Mode Traffic in the system can be determined.

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID
5	Unsigned-8	Number of different publish data commands received from this device
6-7	Unsigned-16	Command Number being published
8-11	Unsigned-32	Number of Publish Data packets received
etc		Repeat bytes 6-11 for each publish data mode command

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid selection
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-127		Undefined

8.5.5 Command 836 Flush Cached Responses for a Device

This is a Gateway Command.

It instructs the Gateway to flush the cached responses that a device participates in.

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID for device to flush all cached responses for

Response Data Bytes

Byte	Format	Description
0-40	Unsigned-40	Unique ID for device to flush all cached responses for

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.5.6 Command 836 Write Update Notification Bit Mask for a Device

This is a Gateway Command.

It registers a client for notification updates.

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Target device Unique ID
5-6	Bits-16	Change notification flags (see HCF Enumeration Table 60)

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Target device Unique ID
5-6	Bits-16	Change notification flags

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-15		Undefined
16	Error	Access Restricted
17-64		Undefined
65	Error	Unknown Unique ID
66	Error	Unknown Notification Flag
67-127		Undefined

8.5.7 Command 838 Read Update Notification Bit Mask for a Device

This is a Gateway Command.

This command asks the Gateway to return the list of update notifications for a Device.

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Target device Unique ID

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Target device Unique ID
5-6	Bits-16	Change notification flags (see HCF Enumeration Table 60)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-64		Undefined
65	Error	Unknown Unique ID
66-127		Undefined

8.5.8 Command 839 Change Notification

This is a Gateway Command.

This request is sent by the Gateway to the Client. The notification lists the changes for the client. Up to 10 change notifications can be included in the response message.

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Target device address

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Target device address
5	Unsigned-8	Number of change notifications
6-7	Unsigned-16	Change notification 1 (cached command number)
		Repeat for up to 10 change notifications

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-64		Undefined
65	Error	Unknown Unique ID
66-127		Undefined

8.5.9 Command 840 Read Network Device's Statistics

This is a Gateway Command.

This command returns the number of graphs, frames, and links that a device has currently active. This is information the Gateway has available and can respond with immediately.

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of the device

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of the device
5-6	Unsigned-16	Number of Graphs active
7-8	Unsigned-16	Number of Frames active
9-10	Unsigned-16	Number of Links active
11	Unsigned-8	Number of Neighbors
12-15	Time	Average communication Latency from the Gateway to this node
16-17	Unsigned-16	Number of Joins
18-20	Date	Date of most recent Join
21-24	Unsigned-32	Time in 1/32 of a ms units from 12:00 am of the Date when the device most recently joined the network
25-28	Unsigned-32	Number of packets generated by this device
29-32	Unsigned-32	Number of packets terminated by this device
33-36	Unsigned-32	Number of Data-Link Layer MIC failures detected
37-40	Unsigned-32	Number of Network Layer (Session) MIC failures detected
41-44	Unsigned-32	Number of CRC Errors detected

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid selection
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-127		Undefined

8.5.10 Command 841 Read Network Device Identity using Nickname

This is a Gateway Command. This command returns the identity information for the indicated device

NOTE This command only needs to be implemented by Gateways.

Request Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname
2-6	Unsigned-40	Unique ID of the device
7-38	Latin-1	Long Tag

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid selection
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-127		Undefined

8.5.11 Command 842 Write Network Device's Scheduling Flags

This is a Gateway Command.

This command allows users to request special consideration for a device when the Network Manager is creating schedules.

This command shall only be supported by Gateways and Network Managers. This information shall not be written to the network device. This command shall not be supported by other network devices (e.g., Field Devices, Adapters).

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of Network ID
5	Bits	Device Scheduling Flags (see HCF Enumeration Table 62)

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of Network ID
5	Bits	Device Scheduling Flags

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too few data byte received
6		Undefined
7	Error	In Write Protect Mode
8	Warning	Unsupported Property Flag detected, Device Property Flags adjusted
9	Error	Invalid Property Flag
10–64		Undefined
65	Error	Unknown Unique ID

Code	Class	Description
66-127		Undefined

8.5.12 Command 843 Read Network Device's Scheduling Flags

This is a Gateway Command.

This command reads the network device properties that a network manager may consider when creating schedules.

This command shall only be supported by Gateways and Network Managers. This information shall not be supported in other network devices (e.g., Field Devices).

Request Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of Network ID

Response Data Bytes

Byte	Format	Description
0-4	Unsigned-40	Unique ID of Network ID
5	Bits	Device Scheduling Flags (see HCF Enumeration Table 62)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid selection
3-4		Undefined
5	Error	Too Few Data Bytes Received
6-127		Undefined

8.5.13 Command 844 Read Network Constraints

This is a Gateway Command.

This command reads the current setting of the Network Management Strategy and the number of Request/Response messages per 10 s.

Request Data Bytes

Byte	Format	Description
None		

Response Data Bytes

Byte	Format	Description
0	Enum-8	Network Optimization Flags (see HCF Enumeration Table 62)
1	Unsigned-8	Number of Request/Response Message Pairs per 10 s

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-127		Undefined

8.5.14 Command 845 Write Network Constraints

This is a Gateway Command.

This command writes the current setting of the Network Management Strategy and the number of Request/Response messages per 10 s. Minimum value for Number of Request/Response Messages per 10 s is 1.

Request Data Bytes

Byte	Format	Description
0	Enum-8	Network Optimization Flags (see HCF Enumeration Table 62)
1	Unsigned-8	Number of Request/Response Messages per 10 s

Response Data Bytes

Byte	Format	Description
0	Enum-8	Network Optimization Flags (see HCF Enumeration Table 62)
1	Unsigned-8	Number of Request/Response Messages per 10 s

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1		Undefined
2	Error	Invalid selection (illegal or unsupported strategy)
3-4		Undefined
5	Error	Too few data bytes received
6		Undefined
7	Error	In Write Protect Mode
8	Warning	Set to nearest value
9-15		Undefined
16	Error	Access restricted
17-127		Undefined

8.6 Network Management Configuration Commands

8.6.1 Command 960 Disconnect Device

This is a Wireless Network Manager Command.

This command allows the network manager to force a device off the network, clear all its network information and rejoin the network.

This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0	Unsigned-8	Reason. (See HCF Enumeration Table 50. Disconnect Cause Codes)

Response Data Bytes

Byte	Format	Description
0	Unsigned-8	Reason. (See HCF Enumeration Table 50. Disconnect Cause Codes)

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-15		Undefined
16	Error	Access Restricted
17-127		Undefined

8.6.2 Command 961 Write Network Key

This is a Wireless Network Manager Command.

This command allows the Network Manager to write the network key on a Network Device.

This command can be truncated after the Key Value. When truncated, the Network Key will become effective immediately.

This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0-15	Unsigned-128	Key value
16-20	Unsigned-40	Execution time for command (ASN). 0 - execute immediately

Response Data Bytes

Byte	Format	Description
0-15	Unsigned-128	Key value
16-20	Unsigned-40	Execution time for command (ASN). 0 - execute immediately

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-15		Undefined
16	Error	Access Restricted
17-64		Undefined
65	Error	Key change failed
66	Error	Invalid execution time
67-127		Undefined

8.6.3 Command 962 Write Device Nickname Address

This is a Wireless Network Manager Command.

This command allows the Network Manager to set a Network Device's Nickname.

This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname

Response Data Bytes

Byte	Format	Description
0-1	Unsigned-16	Nickname

Command-Specific Response Codes

Code	Class	Description
0	Success	No Command-Specific Errors
1-4		Undefined
5	Error	Too Few Data Bytes Received
6-15		Undefined
16	Error	Access Restricted
17-64		Undefined
65	Error	Invalid Nickname
66-127		Undefined

8.6.4 Command 963 Write Session

This is a Wireless Network Manager Command.

This command allows the Network Manager to write the session parameters required to establish a session between the device the message is addressed to and the peer device contained in the request.

This command can be truncated after the Session Key Value. When truncated, the session and the Session Key will become effective immediately.

This command shall be rejected with a response code of 16 (Access Restricted) if the source address is any device other than the Network Manager.

Request Data Bytes

Byte	Format	Description
0	Enum-8	Session type. (See HCF Enumeration Table 48. Session Type Code)
1-2	Unsigned-16	Nickname of peer device
3-4	Unsigned-16	Peer Expanded Device Type Code
5-7	Unsigned-24	Peer Device Id
8-11	Unsigned-32	Peer Nonce counter value
12-27	Unsigned-128	Key value
28	Unsigned-8	Reserved shall be set to 0