STANDARD

WirelessHART Device Specification

HCF_SPEC-290, Revision 1.1

Release Date: 22 May, 2008

Release Date: 22 May, 2008

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

Copyright © 2007 (Rev. 2008) HART® Communication Foundation
This document contains copyrighted material and may not be reproduced in any fashion without the written permission of the HART Communication Foundation.

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

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

http://www.hartcomm.org

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

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

Addendum To
Wireless Devices Specification (HCF_SPEC-290)

May 22, 2008

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

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

Subsection 6.3.1
In Subsection 6.3.1, the second bullet must be modified as follows:

- 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).

Subsection 6.3.2
Subsection 6.3.2, must be modified to clarify the use of Request, Delete, and Write Service as follows:

Burst Messages are used to publish data to applications. In general, the Gateway provides access to the WirelessHART network and caches the published data. The Burst Mode is configured as required to meet the process or plant equipment requirements. This configuration is performed using standard HART procedures (see *Common Practice Command Specification*). For example, Command 108 is used to select the command to be published and Command 109 turns publishing on and off (see Figure 8).

Command 109 is used to start Burst Mode publishing 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 may initiate a Delayed Response (DR) advising the application that it processing the request. The Field Device next issues a "Request Service" (Command 799) to the Network Manager. The Network Manager may answer immediately if it is certain adequate resources can be made available. Alternatively, the Network Manager returns a DR to the Field Device and begins processing the request. Once the device receives the response completing Request Service command transaction it competes its Burst Message Control transaction with the application.

The Network Manager may deny the Request Service or provide it a lower rate. Should this occur, the device must set the "Capacity Denied" status and the "More status available" bit in the Device

Copyright © 2008 HART® Communication Foundation

Status byte if the bandwidth restriction can result in a process upset/disturbance. Under no circumstance shall a device consume more communication bandwidth than allocated by the Network Manager.

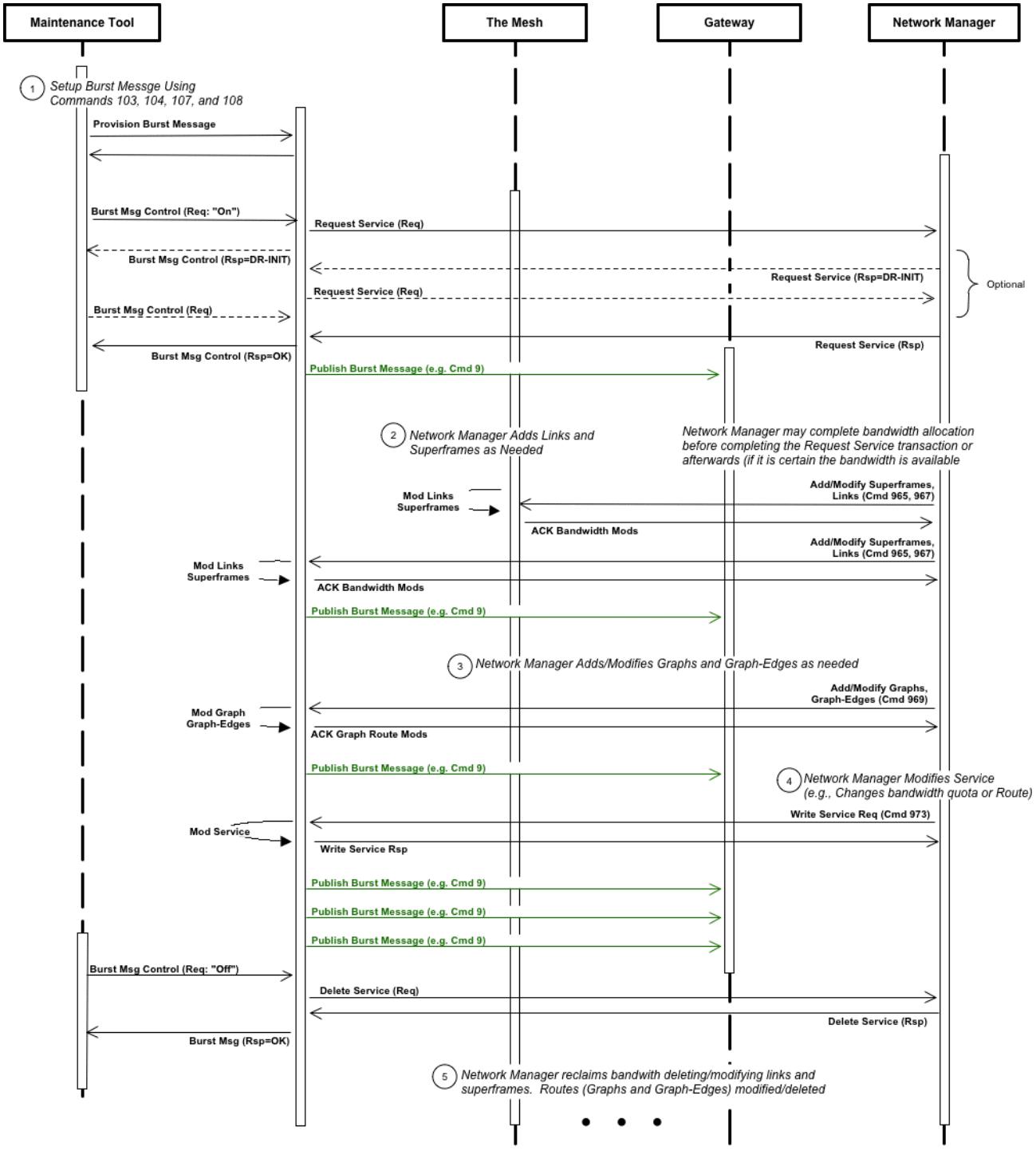


Figure 8. Supporting Burst Mode Data Transfer

Upon receiving the Request Service command, the Network Manager allocates the network bandwidth and routes (see ② in Figure 8) issuing commands to set up Superframes and Links. In most cases, the Network Manager will set up Links in an existing Superframe to satisfy this request. In some cases Routes must be added or modified (see ③ in Figure 8).

Later the Network Manager may choose to issue "Write Service (Command 973) to modify the Service (see ④ in Figure 8). For example, the Network Manager may need to change the route or the bandwidth quota. When "Write Service" is used to reduce previously allocated bandwidth the device must set the "Capacity Denied" status and the "More status available" bit in the Device Status byte if the bandwidth restriction can result in a process upset/disturbance.

Once initiated, the Field Device then published data indefinitely (potentially for years). The Field Device will continue to publish data until it instructed to stop. To request a device to stop publishing data the application issues another Command 109 (see ⑤ in Figure 8), with Burst mode Control set to "Off". The Field Device in-turn will send a "Delete Service" (Command 801) to the Network Manager. The Network Manager will immediately respond and then reconfigure the network thus reclaiming resources accordingly.

Subsection 6.3.3
Subsection 6.3.3, must be modified to clarify the use of Request, and Delete Service as follows:

Block Data Transfer is used for segmented transfer of large data-sets between a device and an application (see *Block Data Transfer Specification*). Data sets can be, for example, tabular data, waveforms, or even firmware upgrades. The Block Transfer can be initiated by the host application or (e.g., periodically) by the device. Figure 9 illustrates a Block Transfer initiated by the device.

Since the device is initiating the transfer, it is responsible for acquiring bandwidth (see ① in Figure 9). Consequently, the device issues a "Request Service" (Command 799) to the Network Manager. In Figure 9, the Network Manager returns a Delayed Response to the Field Device and begins processing the request. Alternatively, the Network Manager may answer immediately if it is certain that resources can be made available

Since Block Transfers are designed to use available bandwidth (rather than requiring a fixed allotment), the Network Manager should always allocate some bandwidth in response to a Block Transfer Service request. However, the Network Manager may deny the Request Service. Should this occur, the device must set the "Block Transfer Pending" status and the "More status available" bit in the Device Status byte. This allows the Gateway to perform the block transfer using its Maintenance Service to the device. Under no circumstance shall a device (or Gateway) consume more communication bandwidth than allocated by the Network Manager.

Upon receiving the Request Service command, the Network Manager allocates the network bandwidth and routes (see ② in Figure 9) issuing commands to set up Superframes and Links. In most cases, the Network Manager will set up Links in an existing Superframe to satisfy this request. In some cases Routes must be added or modified (see ③ in Figure 9). Once the Network Manager completes modifications to the network schedule, it completes the Request Service transaction (see ④ in Figure 9) thus allowing the device to begin the Block Transfer.

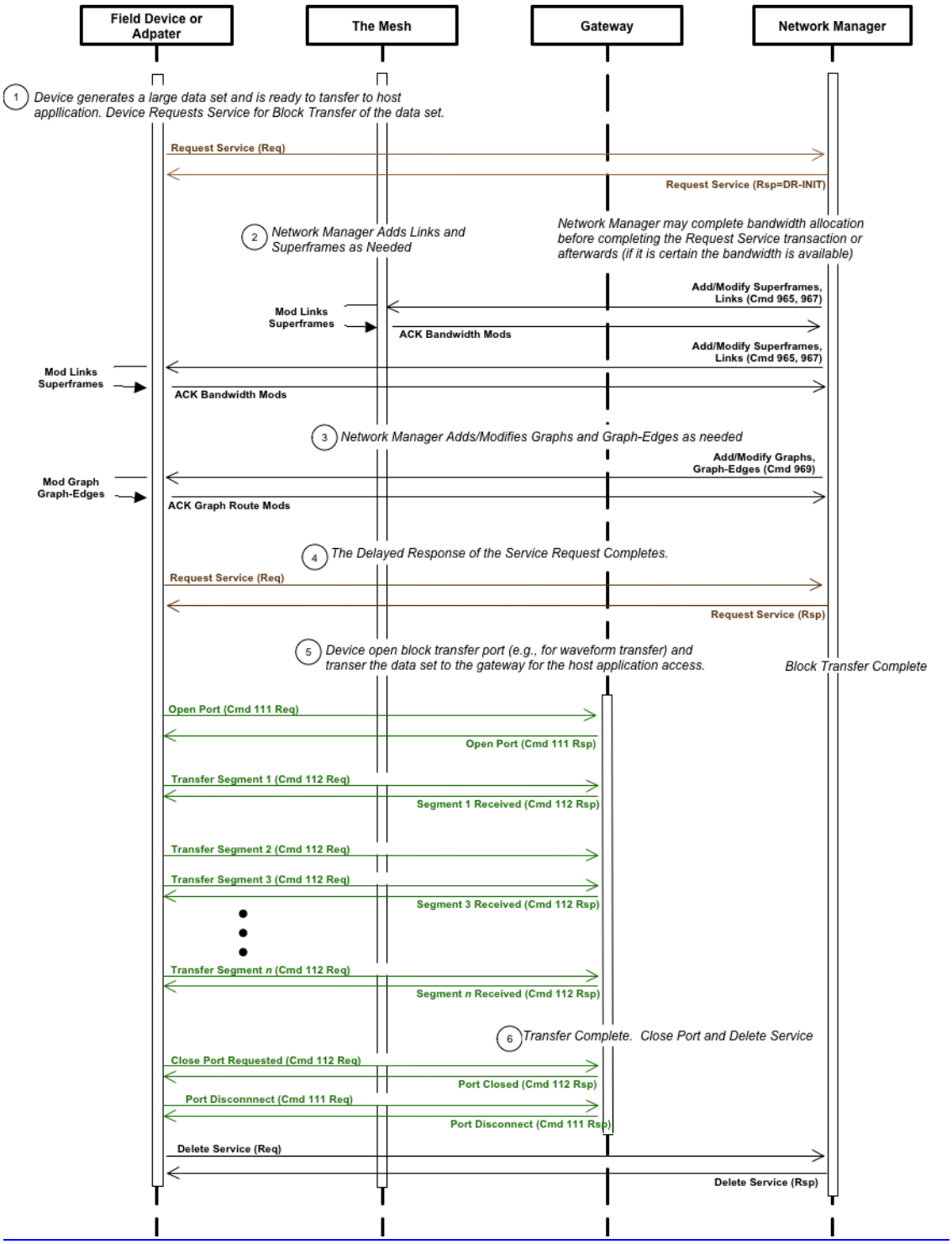


Figure 9. Block Data Transfer (initiated by the Device)

Using Command 111, the device opens a port for communication with a Field Device (see ⑤ in Figure 9). Via the Gateway the host application will complete the port opening sequence by returning its response to Command 111. The port is now open.

Once the port is open, the device and application exchange Command 112 messages to move the data. Unlike operation using the Token-Passing Data-Link Layer, the principal data source (e.g, the device) may issue multiple Command 111 packets without waiting peer acknowledgement. When the transmission is done, the device will complete the block data transfer by issuing a Command 111 request to close the port (see ⑥ in Figure 9). The Field Device will send a "Delete 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.

Block transfer may also be initiated by the Gateway. In this case, the Network Manager must use Write Timetable (Command 973) to specify the bandwidth quota the field device may utilize while performing the block transfer (see xxxx).

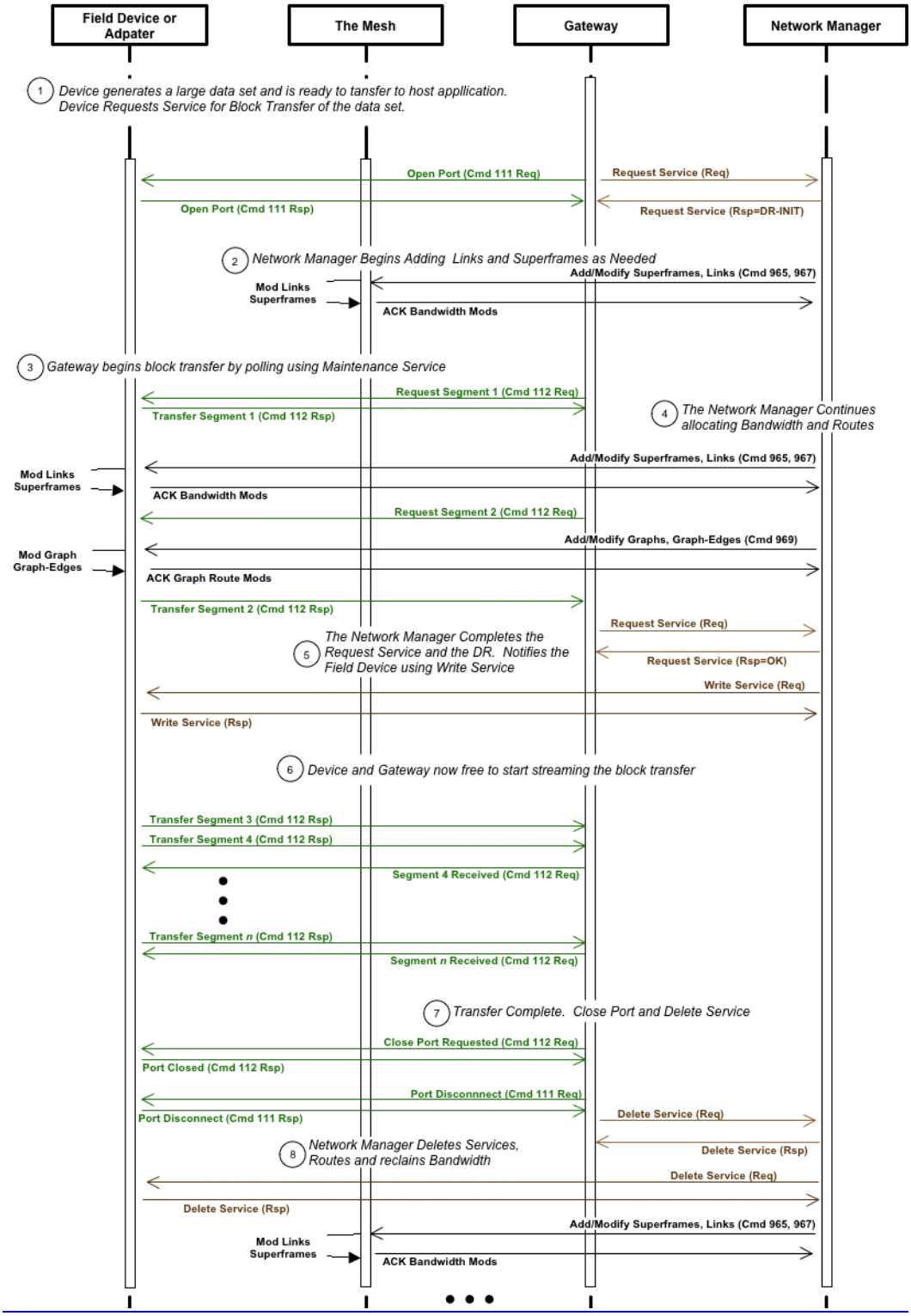
Note: While the details of the communication between the Gateway and Network Manager are not governed the HART Specifications, the Request and Delete Timetable commands are used to illustrate the sequence in xxxx.

Since the Gateway is initiating the transfer, it is responsible for acquiring bandwidth (see ① in xxxx). Initiating the bandwidth can be performed simultaneously with opening the port using Command 111. Consequently, the Gateway requests a Timetable from the Network Manager. This request maybe delayed while the Network Manager configures the network for the block transfer.

While the Block Transfer Timetable is being arranged (see ② in xxxx), the Gateway begins issuing Command 112 requests using the existing Maintenance Timetable (see ③ in xxxx). The acknowledged Transport Layer pipe is used as is normal for Maintenance traffic. The Device responds (as usual) to this request/response traffic. Meanwhile the Network Manager continues configuring the Block Transfer Timetable (see ④ in xxxx).

Upon receiving the Request Timetable command, the Network Manager allocates the network bandwidth and route issuing commands to set up Superframes and Links. In most cases, the Network Manager will set up Links in an existing Superframe to satisfy this request. In some cases Routes must be added or modified. Once the Network Manager completes modifications to the network schedule, it completes the Request Timetable transaction (see ⑤ in xxxx) . When the Network Manager completes configuring the Block Transfer Timetable it issue the Write Timetable command specifying the bandwidth quota available to the device. Under no circumstance shall a device (or Gateway) consume more communication bandwidth than allocated by the Network Manager.

Once the Block Transfer Timetable is established the principal data source (e.g., the device) may issue multiple Command 111 packets without waiting peer acknowledgement (see ⑥ in xxxx). When the transmission is done, the device will complete the block data transfer by issuing a Command 111 request to close the port (see ⑦ in xxxx). The Gateway then asks the Network Manager to delete the Timetable (see ⑧ in xxxx). The Network Manager will immediately answer this request and transmit the Delete Timetable to the Device. Then the Network Manager reconfigures the network accordingly and reclaims the bandwidth.



xxxx <new figure> Block Data Transfer (Initiated by Gateway)

Subsection 6.3.4

Subsection 6.3.4, Nonce Counter History must be added as follows:

The field device shall maintain a NonceCounterHistory array at least 32 bits long. As specified in the Network Management Specification, the PeerNonceCounter is largest nonce counter value validated by the Field Device. As it increases the NonceCounterHistory is shifted. Normally the bits shifted out are set (i.e., 1). When a reset bit (0) is shifted out the device must increment the Nonce Counter History Underflow statistic (See Command 779).

Subsection 7.1

The requirements for supporting time synchronization and all standardized commands and procedures must be clarified as follows:

In addition to Universal and Common Practice Commands, the Field Device must support the standardized commands and procedures defined for WirelessHART devices. This includes the requirements for polling by tag. In other words, if the Adapter receives a broadcast Command 11 or 21 it must forward it to its sub-devices and respond to the Gateway accordingly.

If the sub-device is HART 7 (or later) then the adapter must be capable of synchronizing the sub-device to network time at least once every 24 hours. This must be accomplished using Commands 89 and 90 if the sub-device supports them.

Command 59 is required for both wireless Field Devices and Adapters. Consequently, Table 3 and 4 must be corrected as follows:

Table 3. Required Common Practice Commands for Adapters

Common Practice

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
74	Read I/O System Capabilities
75	Poll Sub-Device
77	Send Command to Sub-Device
78	Read Aggregated Commands
84	Read Sub-Device Identity Summary
85	Read I/O Channel Statistics
86	Read Sub-Device Statistics
87	Write I/O System Master Mode
88	Write I/O System Retry Count
90	Read Real-Time Clock
95	Read Device Communications Statistics
101	Read Sub-device to Burst Message Map (see Subsection 6.3.2)

Cmd	Description
102	Map Sub-device to Burst Message
103	Write Burst Period
104	Write Burst Trigger
105	Read Burst Mode Configuration
106	Flush Delayed Response Buffers
107	Write Burst Device Variables
108	Write Burst Mode Command Number
109	Burst Mode Control
111	Transfer Service Control (on behalf of the devices)
112	Transfer Service
115	Read Event Notification Summary
116	Write Event Notification Bit Mask
117	Write Event Notification Timing
118	Event Notification Control
119	Acknowledge Event Notification

While not required, an Adapter may provide an analog interface to measure or control the loop current. When this is the case Commands 35, 40, 45, 46, 54, 79 must be supported to allow access to and control of the loop current. The loop current must be mapped to a Device Variable (see *Command Summary Specification*).

Table 4. Recommended Common Practice Commands for Adapters

Common Practice

Cmd	Description
71	Lock Device
72	Squawk
73	Find Device
76	Read Lock Device State

Cmd	Description
94	Read I/O System Client-Side Statistics
512	Read Country Code
513	Write Country Code

If the Adapter provides an analog interface to measure or control the loop current, Commands 33, 34, 50, 55, 80-83 should be supported to allow access to and control of the loop current.

In Table 5, the requirements for Adapters to support Burst and Event Messages must be modified as follows:

Table 5. 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 Burst (Data) Messages	2 + 2*Number of Sub-Devices
Minimum Number of Burst (Event) Messages	1 + Number of Sub-Devices

Subsection 7.2.1

Collision avoidance must operate the same irrespective of whether the Adapter is configured as a primary or secondary master. Consequently, Subsection 7.2.1 must be modified to read as follows:

By default, the Adapter is a primary master and communicates continuously adhering to the requirements in the *Token-Passing Data-Link Layer Specification*. It may also be configured as a secondary master.

The Adapter must be prepared to defer to a master with the same address (e.g., a Handheld). In other words, if configured as a primary (or secondary) master and another primary (or secondary) master is detected; the Adapter must cease issuing master requests and set the "Duplicate Master Detected" status in Command 48 (see Common Table 31) and set "More Status Available" status.

Note: Whenever possible, one of the connected devices must be configured into burst mode.

When a conflict with another master occurs the Adapter must cease initiating transactions until that master's communication is not detected for 4 times the Link Quiet Time (RT1). The Link Quiet Time for an Adapter shall be RT1 (Secondary) plus RT2.

Subsection 7.3.1
In Subsection 7.3.1, the second bullet must be modified as follows:

- 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).

Subsection 7.3.2
Delayed responses from field devices (for example while performing calibration) may take a relatively long time. Consequently, the last paragraph in Subsection 7.3.2 must be modified as follows:

Otherwise, the response from the field sub-device is considered valid and will be returned to the Gateway. If the allowed number of retries is exceeded, DR_DEAD will be returned to the Gateway. A Busy or a DR response indicates communication is successful but the device is unable to complete the request. Upon receiving a busy or delayed response the adapter must continue to perform retries to the sub-device to complete the transaction. See Command Response Code Specification for more information.

Subsection 7.3.4
Subsection 7.3.4, Nonce Counter History must be added as follows:

The adapter shall maintain a NonceCounterHistory array at least 32 bits long. As specified in the Network Management Specification, the PeerNonceCounter is largest nonce counter value validated by the adapter. As it increases the NonceCounterHistory is shifted. Normally the bits shifted out are set (i.e., 1). When a reset bit (0) is shifted out the device must increment the Nonce Counter History Underflow statistic (See Command 779).

Subsection 8.2.9
Subsection 8.2.9 shall be deleted.

Subsection 8.2.11
The specification must clearly state that communications with sub-devices must be transparent to the host application. The following paragraph must be added to the Subsection 8.2.11:

The Gateway shall cache status, process data and other static device data. Communication with sub-devices must be transparent. In other words, commands addressed to sub-devices must be automatically wrapped in a Command 77 request/response. The response to the host application shall be the payload in the Command 77 response from the Adapter.

Subsection 8.3.8 - Adapters and Sub-Devices
The specifications for Gateway management of sub-devices must be modified to support HART 5 devices as follows:

If the Network Device is an Adapter (i.e. a protocol bridge device) then the Gateway must determine the number of sub-devices connected to the Adapter and identify each of the sub-devices (see the *Common Practice Command Specification*). The Gateway determines which devices are Adapters by examining byte 8 (Flags), bit 2 (Protocol Bridge Device) and the Device Profile in the Identity Command (Command 0).

The Gateway sends each sub-device a Command 0 and Command 20. The response messages for each sub-device are cached by the Gateway. For HART 5 devices connected to the Adapter, Long Tag shall be simulated using the "Message" attribute as read from Command 12.

Subsection 8.6.4
The specification must clearly state that communications with sub-devices must be transparent to the host application. The following paragraph must be added to the Subsection 8.6.4:

Communication with sub-devices must be transparent. In other words, commands addressed to sub-devices must be automatically wrapped in a Command 77 request/response. The response to the host application shall be the payload in the Command 77 response from the Adapter.

Subsection 8.6.7/Table 8
The minimum size of the NonceCounterHistory bit-array is proportional to the size of the network. Consequently a new entry in Table 8 must be added as follows:

Parameter	Tiny (10 devices)	Small Gateway (50 devices)	Large Gateway (250 devices)
Minimum NonceCounterHistory length	32bits	128bits	256bits

In addition the following paragraph must be added to the end of the section:

The Gateway shall maintain a NonceCounterHistory array at least as long as indicated in Table 8. For network sizes between the sizes indicated, the length chosen shall be that of the next larger network. For example, for a network of 100 devices the Gateway must support a history length of at least 256bits. For networks larger than 250 devices the history length should be increased as needed to ensure valid packets are retained.

As specified in the Network Management Specification, the PeerNonceCounter is largest nonce counter value validated by the Gateway. As it increases the NonceCounterHistory is shifted. Normally the bits shifted out are set (i.e., 1). When a reset bit (0) is shifted out the Gateway must increment the Nonce Counter History Underflow statistic. Command 840 returns both the underflow count from the network device and the Gateway. The statistic returned by the Gateway includes the Nonce Counter History Underflow detected by the Network Manager.

Subsection 8.6.8/Table 9

Gateway use notification interface (see Commands 837-840) to signal host applications. Consequently, the Common Practice publish and event commands are redundant. In addition the requirement for Command 78 "Write Time of Day" is actually Command 89 "Set Real-Time Clock" Consequently, Table 9 must be updated to read as follows:

Table 9. Required Commands

Common Practice 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
74	Read I/O System Capabilities
75	Poll Sub-Device
77	Send Command to Sub-Device
84	Read Sub-Device Identity Summary

Cmd	Description
85	Read I/O Channel Statistics
86	Read Sub-Device Statistics
87	Write I/O System Master Mode
88	Write I/O System Retry Count
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 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 Burst Mode 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

(New Table) Recommended Commands

Common Practice Commands

Cmd	Description
71	Lock Device
76	Read Lock Device State

Cmd	Description
512	Read Country Code
513	Write Country Code

WirelessHART Commands

Cmd	Description
793	Write UTC Time Mapping

Subsection 9.2.1

Subsection 9.2.1, Latency and Bandwidth Management must be added as follows:

The Network Manager must manage routes and communication bandwidth to minimize latency and out of order packet arrival. To tolerate out of order packet delivery the Network Manager's NonceCounterHistory array must be at least as long as:

- *32bits for Tiny networks (0-10 devices);*
- *128bits for Small networks (11-50 devices); or*
- *256bits for Large networks (51-250 devices).*

For networks larger than 250 devices the history length should be increased as needed.

As specified in the Network Management Specification, the PeerNonceCounter is largest nonce counter value validated by the Network Manager. As it increases the NonceCounterHistory is shifted. Normally the bits shifted out are set (i.e., 1). When a reset bit (0) is shifted out the Network Manager must increment the Nonce Counter History Underflow statistic and provide it to the Gateway for reporting.

Table of Contents

Table of Figures

List of Tables

Preface

This preface is included for informational purposes only.

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

Starting in late 2005, the HCF began evaluating wireless technology and developing a wireless alternative to traditional wired HART. This specification is a result of that development effort. WirelessHART™ provides technology for devices within the WirelessHART network and for host application accessing the WirelessHART network. The standard also provides support for existing HART field devices and handhelds.

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

Preserve and enhance industry's existing investment in HART Technology

Leverage established technologies, standards, and practices to rapidly develop backward-compatible WirelessHART Standards.

Maximize coexistence by ensuring reliable WirelessHART communications while minimizing interference to other wireless technologies.

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

The following people actively served as members or made significant contributions to the WirelessHART Device Working Group during the creation of this document:

Tomas Lennvall	ABB
Mark Nixon	Emerson Process Management
Robin Pramanik	Siemens AG
Wally Pratt	HART Communication Foundation
Eric Rotvold	Rosemount Inc.
Yuri Zats	DUST Networks

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

Introduction

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

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

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

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

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

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

WirelessHART™ is a secure, networking technology operating in the 2.4GHz ISM radio band. As currently envisioned, WirelessHART utilizes IEEE 802.15.4 compatible DSSS radios with channel hopping on a packet by packet basis. WirelessHART communication is arbitrated using WirelessHART Network to schedule link activity. A given WirelessHART Network slot may be dedicated to communication between a network pair or a slot may support Slotted-Aloha shared communication access.

HART is a master-slave protocol and is loosely organized around the ISO/OSI 7-layer model for communications protocols (see Figure 1). With the introduction of wireless technology to HART, a second Physical and Data Link Layer are supported: the wireless mesh (refer to the *TDMA Data Link Layer Specification*). WirelessHART continues to support the common HART Application Layer. WirelessHART also significantly adds to and expands the capabilities of HART (refer to the *Common Practice Command Specification)*. In addition, since WirelessHART allows deployment of

mesh topologies a significant network layer is now specified (refer to the *Network Management Specification*).

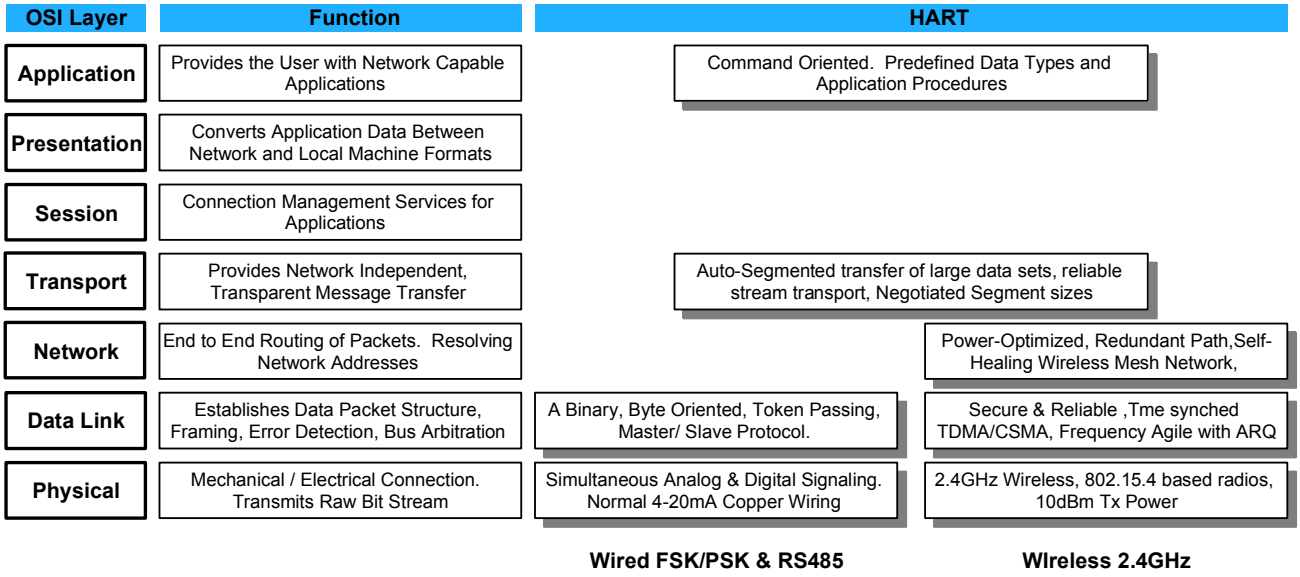


Figure 1. OSI 7-Layer Model

The WirelessHART Architecture is designed to be an easy to use, reliable and inexpensive wireless mesh sensor protocol. WirelessHART equipment consists of core mandatory capabilities that allow equivalent device types to be exchanged without compromising system operation. To this end the majority of WirelessHART requirements is mandatory and must be universally supported.

Furthermore, WirelessHART is backward compatible to HART core technology such as the Device Description Language. All HART devices (e.g., network managers, gateways, field devices, etc) shall support DDL.

This specification walks through the devices that make up the WirelessHART architecture. The document begins with an overview that presents several possible topologies for deploying WirelessHART. The document then walks through Field Devices, Adapters, Gateways, the Network Manager, and Handhelds. This document builds on and references materials in the *TDMA Data Link Layer Specification* (HCF_SPEC-75), the *Network Management Specification* (HCF_SPEC-85), the *Common Practice Command Specification* (HCF_SPEC-151) and the *Wireless Commands Specification* (HCF_SPEC-155).

1. SCOPE

This document is an Application Layer specification and, as a result, builds on the Application Layer requirements found in the *Command Summary Specification* (HCF_SPEC-99). Conformance to the *Universal Command Specification* requires *Command Summary Specification* conformance as a prerequisite.

This specification defines compliance requirements for specific WirelessHART device types. For compliance purposes, a WirelessHART product shall be classified as one of five different product types (see Figure 2). These product types are:

- **Field Devices** are mounted in the process and must 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.

- **Adapters** connect to existing HART compatible field devices and enable communication to them via a WirelessHART Network. Adapters must route messages to and from their sub-devices and other devices in the network.

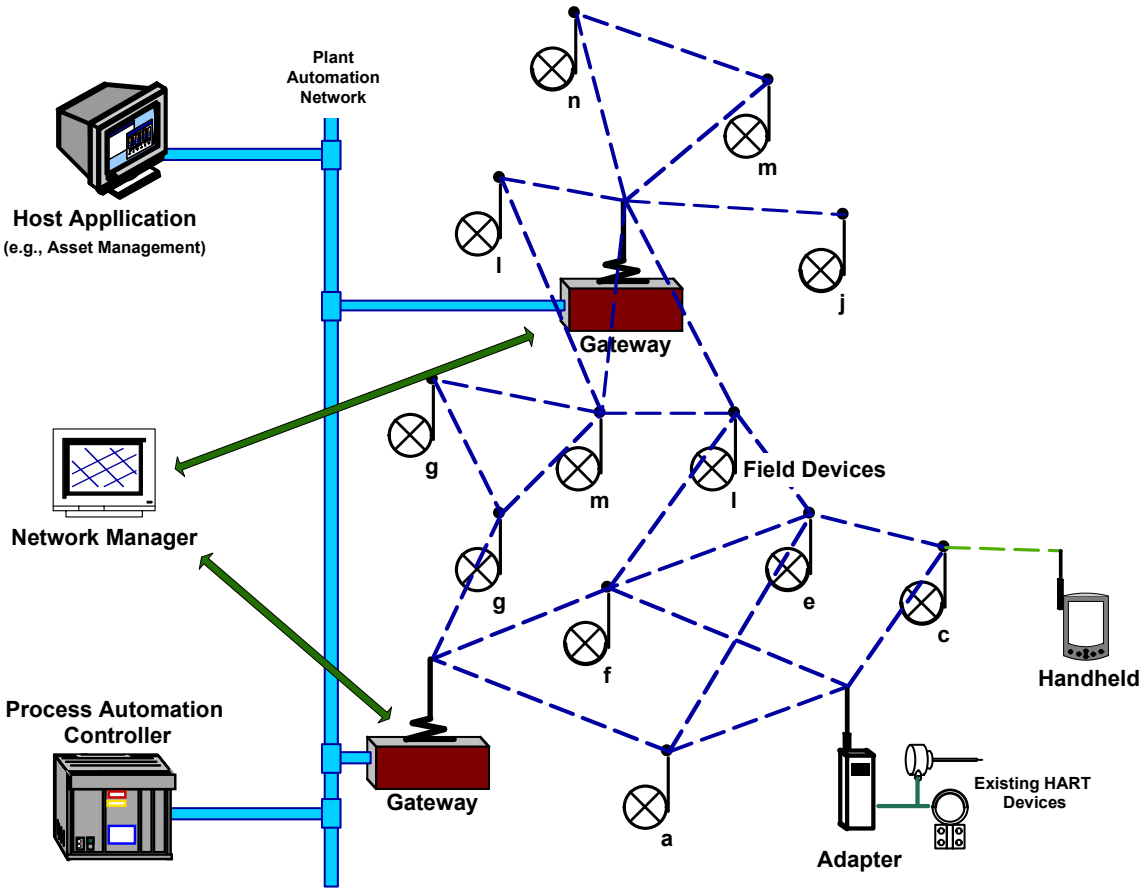


Figure 2. WirelessHART Elements of a WirelessHART Installation.

- 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 fall 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.

2. REFERENCES

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

HART Field Communications Protocol Specification. HCF_SPEC-12.

TDMA Data-Link Layer Specification. HCF_SPEC-75

Network Management Specification. HCF_SPEC-85

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

Command Summary Specification. HCF_SPEC-99

Command Practice Command Specification. HCF_SPEC-151

Wireless Command Specification. HCF_SPEC-155

Block Data Transfer Specification. HCF_SPEC-190

2.1 Related HART Documents

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

HART Field Communications Protocol Specification. HCF_SPEC-12

Token-Passing Data Link Layer Specification. HCF_SPEC-81

Command Summary Specification. HCF_SPEC-99

Universal Command Specification. HCF_SPEC-127

Common Tables Specification. HCF_SPEC-183

Block Data Transfer Specification. HCF_SPEC-190

Command Response Code Specification. HCF_SPEC-307

WirelessHART User Guide. HCF_LIT-84

Coexistence Test Plan. HCF_LIT-85

2.2 Related Documents

The following are applicable IEEE documents:

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

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

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

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

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

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

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

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

The following reference provides a general overview of UTF-8.

Wikipedia contributors, "UTF-8," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=UTF-8 (accessed August 30, 2007)

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

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

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

3. DEFINITIONS

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

Acknowledge	Positive response upon the successful reception and acceptance of a data packet. In WirelessHART it further assumes that the packet has been successfully forwarded to the next hop. Data is delivered on a hop-by-hop and an end-to-end basis.
Broadcast	The sending of packets to all Network Devices that overhear the transmission.
Byte	8-bits. Sometimes called an Octet.
Channel	RF frequency band used to transmit a modulated signal carrying packets.
Channel Hopping	Regular change of transmit / receive frequency to combat interference and fades.
Channel Blacklisting	A method of eliminating an RF channel from usage.
Channel Offset	A link-specific value provided by the Network Manager that is used to calculate the channel to use when channel hopping.
Clear Channel Assessment	Clear Channel Assessment (CCA) is used to avoid transmitting a packet while the RF channel is in use. CCA is performed by listening to the channel prior to beginning a transmission.
Coexistence	Coexistence is the ability of one system to perform a task in a given shared environment in which other systems have an ability to perform their tasks and may or may not be using the same set of rules (IEEE).
Connection	A data structure associated with graph routing that contains an ordered pair of Network Devices.
Data Link Layer	Layer 2 in the OSI 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.
Device Id	A 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.
Discovery	A method to locate or identify WirelessHART Gateways without human interaction.
Frequency Channels	The allocation of the frequency spectrum in a given frequency range.
Gateway	A Network Device containing at least one host interface (such as serial or Ethernet), acting as ingress or an egress point.

Graph	A routing structure that forms a directed end-to-end connection between Network Devices.
Handheld	A host application residing on a portable device.
Hop	The 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.
Interoperability	Interoperability is the ability for like devices from different manufacturers to work together in a system and be substituted one for another without loss of functionality at the host system level.
Join	Process by which a Network Device establishes a connection to the Network Manager.
Latency	The time it takes for a packet to cross a network connection, from sender to receiver. Latency specifications shall (unless otherwise noted) represent a 2-sigma value. i.e., the latency shall be achieved 90% of the time.
Lease	A 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.
Link	The full communication specification between adjacent devices in the network, i.e., 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.
Neighbor	Adjacent devices in the network that have RF connectivity.
Network Manager	The application responsible for configuration of the network; scheduling communication between neighbors; management of the routing tables and monitoring and reporting the health of the network.
Network Device	A device with a direct Physical Layer connection to the network. Each network device (e.g., field device or gateway) has a HART Unique Address that is used in communication with the device. Network Devices include Field Devices, Access Points (i.e. Gateways), Adapters, and Handhelds.
Node	see Network Device
Nonce	A 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.
Packet	A generic reference to the set of data communicated across a network

Physical Layer	Layer 1 in the OSI model. The Physical Layer is responsible for transmission of the raw bit stream and defines the mechanical and electrical connections and signaling parameters for devices.
Receiver Sensitivity	The minimum input signal required to produce a packet error rate of less than 10E-2.[3]
Security Manager	An application that manages the Network Device's security resources and monitors the status of the network security.
Service Session	An agreement between a Client and a WirelessHART Gateway that services shall be provided to the Client by the Gateway.
Slot	A fixed time interval that may be used for communication between neighbors.
Superframe	A collection of slots repeating at a constant rate. Each slot may have a link associated with it.
Throughput	The effective data transfer rate of the network.
Time Sequence Diagram	A diagram used to illustrate the interrelationship between the Protocol services. The protocol layer of interest and the lower, intervening layers are treated as a "black box". The internal workings of these layers are not shown on this diagram. The time sequence diagram shows the interactions between the service primitives over time. Sometimes referred to as a Message Sequence Diagram.
UTF-8	UTF-8 (8-bit UCS/Unicode Transformation Format) is a variable-length character encoding for Unicode. All XML described in this specification is encoded using UTF-8 characters.
Unicast	The sending of a packet to a single node in the network.
Physical Layer	Layer 1 in the OSI model. The Physical Layer is responsible for transmission of the raw bit stream and defines the mechanical and electrical connections and signaling parameters for devices.

3 IEEE Std 802.15.4-2006 defines the conditions for minimum Receiver Sensitivity in section 6.1.7 Table 4 and that is for a PER of 1%.

4. SYMBOLS/ABBREVIATIONS

All Symbols and Abbreviations used in this specification are listed in this section.

ACK	See Acknowledge
ASN	Absolute Slot Number
CCA	**C**lear **C**hannel **A**ssessment
DLL	See **D**ata-**L**ink **L**ayer
DRM	Delayed Response Mechanism
DSSS	**D**irect **S**equence **S**pread **S**pectrum
FTA	**F**ield **T**ermination **A**ssembly (as referenced in Figure 6)
PER	**P**acket **E**rror **R**ate
PHY	See Physical Layer
RSL	**R**eceived **S**ignal **L**evel (in dB; one byte signed).
STX	**S**tart of a **T**ransaction. An STX is the start of the transmission (Tx) of the data packet between nodes. A transaction typically consists of an Tx and a corresponding ACK or NACK response.
WHA	WirelessHART Adapter
WHD	WirelessHART Device

5. OVERVIEW

HART enables communication with smart process instrumentation and controls and supports both wired and wireless networking technologies. Wired communication supports both point-to-point and multidrop topologies.

WirelessHART is made up of several network components. The section begins with a discussion of these components and then presents several possible topologies.

5.1 WirelessHART Network Components

This diagram below will be used to discuss WirelessHART network components. In the diagram 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.

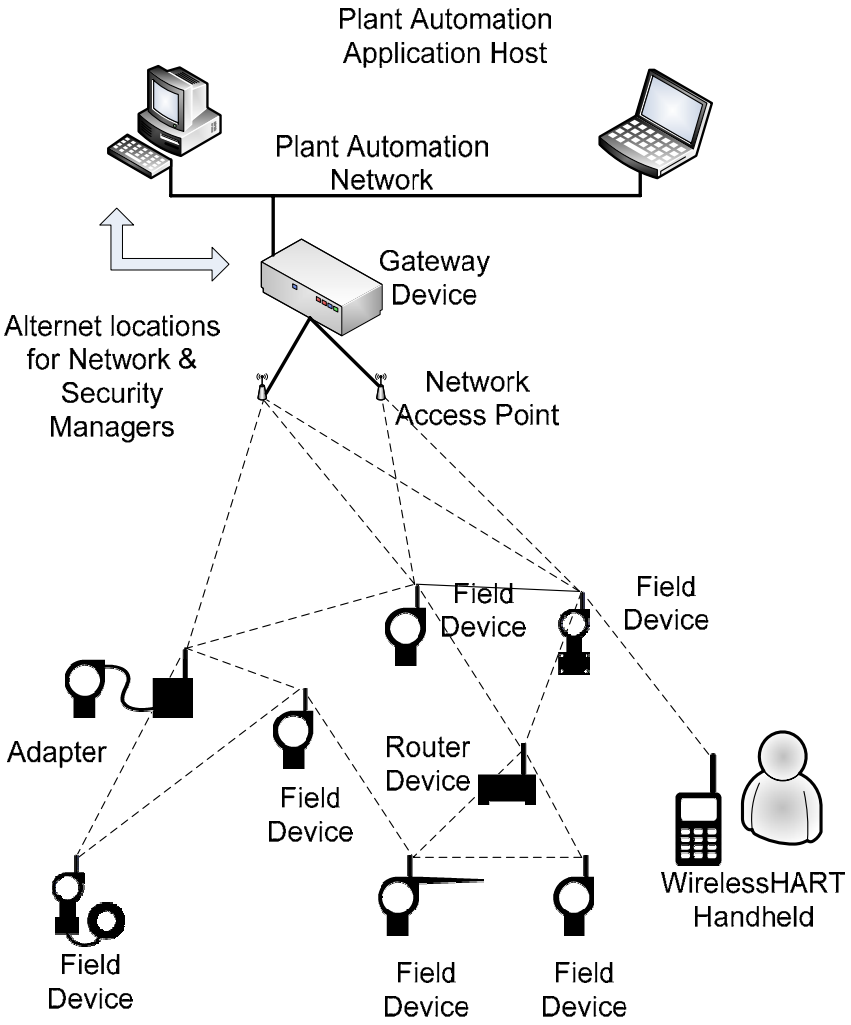


Figure 3. WirelessHART Standalone Gateway.

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 must 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. (See the *TDMA Data-Link Layer Specification*)

5.1.1 Field Device

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

5.1.2 Adapter

An *Adapter Device* is a Network Device that connects wired HART Devices into the WirelessHART Network. An Adapter uses internal routing tables to coordinate traffic flow between the WirelessHART Network and wired Devices. An Adapter is not directly connected to the process.

5.1.3 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.

5.1.4 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.

5.1.5 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 must support routing. However, it may be beneficial (e.g., to extend the Network, or to save the power of a Field Devices 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.

5.1.6 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:

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.

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.

5.1.7 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. The Network Manager is described in detail in this section.

5.2 Network Topologies

5.2.1 WirelessHART Gateway in a PC Card

In the next scenario the WirelessHART Gateway is built into a PC Card. This topology could also be used to support higher level applications such as an asset management application. The primary and secondary measurements, alarms, etc could also be accessed through the Gateway Physical Access Point and processed locally or transmitted over some other network to other plant applications.

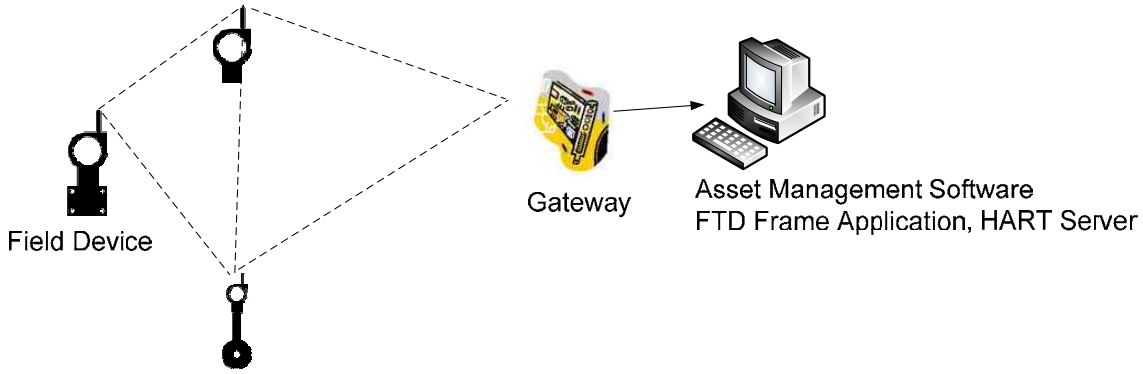


Figure 4. WirelessHART Gateway is Plug-in PC Card

5.2.2 WirelessHART Gateway as part of IO subsystem

In this next scenario, the WirelessHART Gateway is built into the IO subsystem of a larger system. This topology could be used to provide IO measurements for monitoring and control applications in PLC-based , DCS-based systems or SCADA-based systems. Higher level applications, such as asset management applications, can also make use of this topology by tunneling their HART commands through the control network.

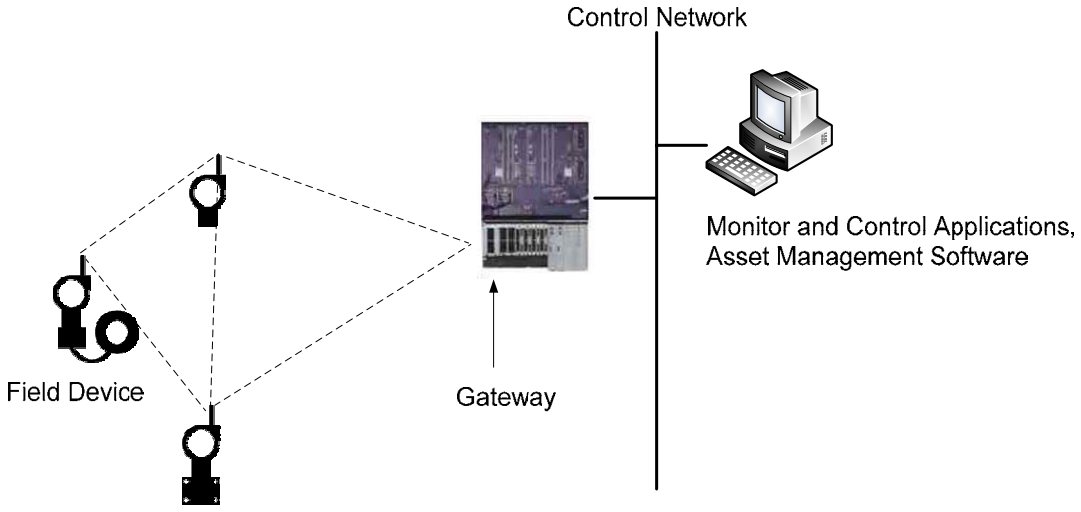


Figure 5. WirelessHART Gateway as part of IO System

5.2.3 Supporting Installed Base

The WirelessHART technology can also work in existing installed based installations. The following drawing illustrates an approach to support this requirement.

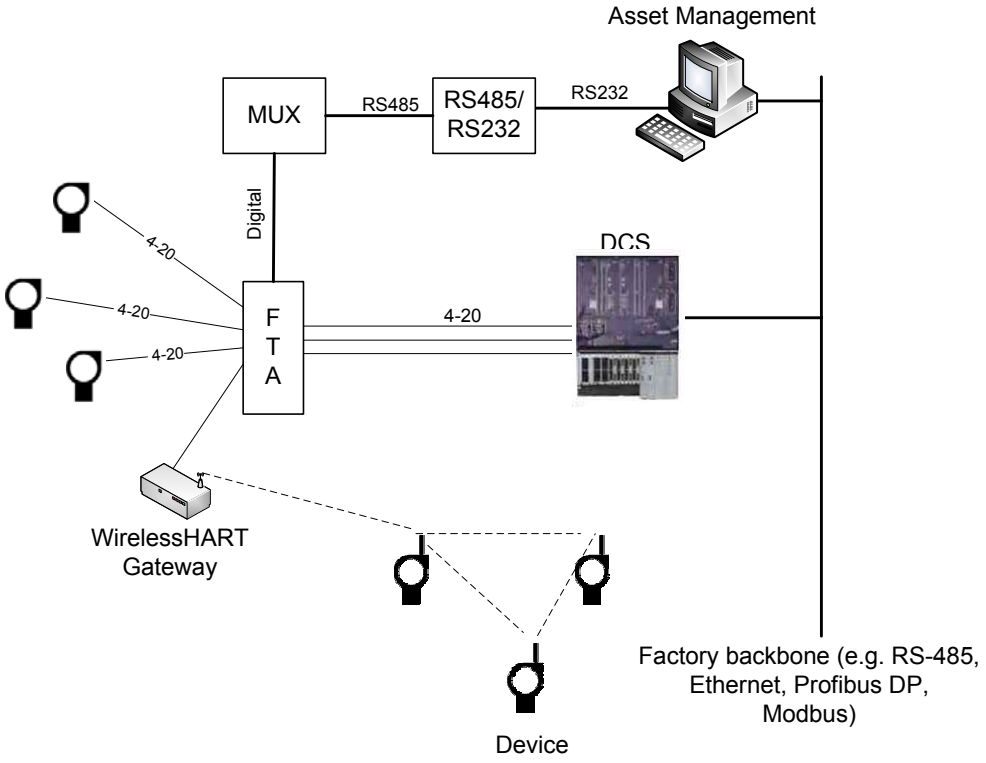


Figure 6. WirelessHART Network and Legacy System

The connection between the WirelessHART Gateway and the FTA[4] is through a RS485 connection. In this case, the WirelessHART Gateway handles commands such as 0-3, 11 and 13. Other commands are passed through to the device as HART commands.

4 FTA stands for Field Termination Assembly.

5.2.4 Adding WirelessHART to existing HART Field Devices

In another scenario, the Field Device is retrofitted with a WirelessHART Adapter.

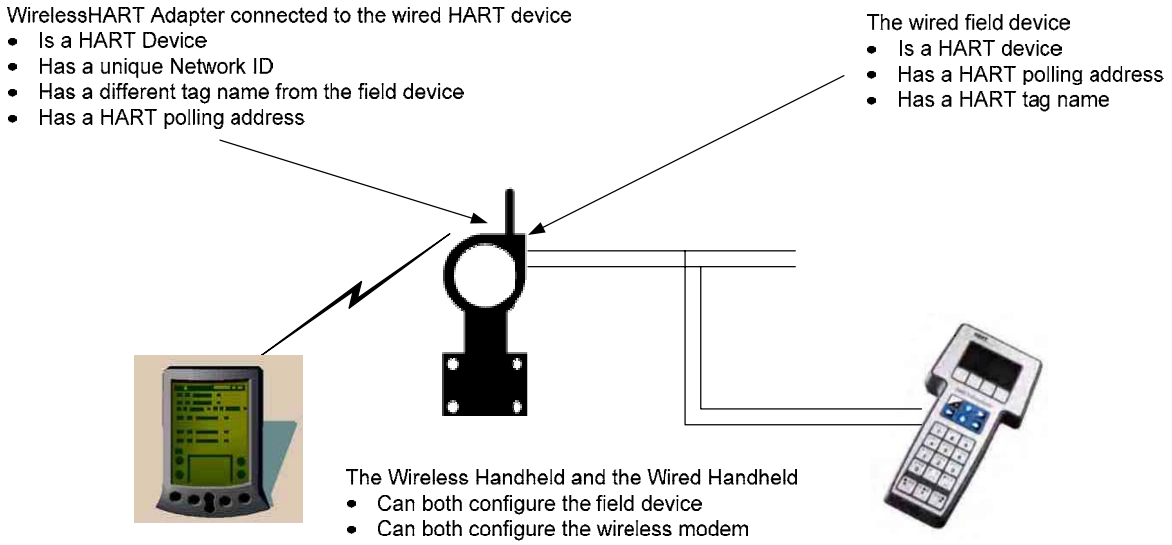


Figure 7. Retrofit of the legacy transmitter with a Wireless Adapter

The wired Handheld connects to the WirelessHART Device through the FSK modem that is installed on every device.

6. WIRELESSHART FIELD DEVICES

The most common type of Network Device is a Field Device. A WirelessHART Field Device is a Network Device that combines wireless communications with traditional HART field device functions. The Field Device may be line, loop, or battery powered or they may be powered in some other fashion. A Field Device is connected the Process or Plant Equipment. Field Devices may or may not support traditional current loop signaling. The WirelessHART field device must have a maintenance port but may not have a HART permanently wired connection to the Process Automation System.

6.1 General Requirements

All Field Devices must support all Universal Commands. In addition, all Field Devices must support the Common Practice Commands found in Table 1 and should support those in Table 2.

Table 1. Mandatory Common Practice Commands for Field Devices

Common Practice

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 Burst Period
104	Write Burst Trigger

Cmd	Description
105	Read Burst Mode Configuration
106	Flush Delayed Response Buffers
107	Write Burst Device Variables
108	Write Burst Mode Command Number
109	Burst Mode 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

Note: While not required, a Field Device may provide an analog interface to measure or control the loop current. When this is the case Commands 35, 40, 45, 46 must be supported to allow access to and control of the loop current (see Command Summary Specification).

Table 2. Recommended Common Practice Commands for Field Devices

Common Practice

Cmd	Description
52	Set Device Variable Zero
53	Write Device Variable Units
55	Write Device Variable Damping Value
71	Lock Device
72	Squawk
73	Find Device
76	Read Lock Device State
80	Read Device Variable Trim Points
81	Read Device Variable Trim Guidelines
82	Write Device Variable Trim Point

Cmd	Description
83	Reset Device Variable Trim
91	Read Trend Configuration
92	Write Trend Configuration
93	Read Trend
95	Read Device Communications Statistics
120	Configure Synchronous Sampling
121	Configure Delayed Command Execution
111	Transfer Service Control
112	Transfer Service

In addition to Universal and Common Practice Commands, the Field Device must support the standardized commands and procedures defined for WirelessHART devices.

6.2 Maintenance Port

All Field Devices must provide a maintenance port that complies with the requirements in the Token-Passing Data Link Layer Specification and support at least one of the Physical Layers that it specifies. This interface is used for maintenance (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 must be available via the maintenance port. When requested, the Field Device answers Identity Commands normally thus allowing the DD-Enabled hosts to load the Field Device's DD.

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

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

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

- Must appear to be a multi-dropped slave device. By default, the Field Device should be configured for polling address 1.
- Must have a 500Ohm input impedance simplifying the direct connection of legacy maintenance tools and applications.
- Should not support burst mode.
- Must not be permanently wired.
- Must be clearly labeled as he maintenance port.

6.3 WirelessHART Interface

The Field Device is a WirelessHART device and must adhere to all WirelessHART specifications. Three key Field Device features include:

- Support for network services; and
- Support for Burst Mode Data Transfer; and
- Support for Block Data Transfer.

In the sections below, network services are described in the context in which they are used.

6.3.1 Timing requirements

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

- 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 first slot following the slot the PDU was received in.
- 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 must not be generated until 75% of the Transport Layer maxReplyTime has elapsed.

6.3.2 Burst Mode Operation

Burst Messages are used to publish data to applications. In general, the Gateway provides access to the WirelessHART network and caches the published data. The Burst Mode is configured as required to meet the process or plant equipment requirements. This configuration is performed using standard HART procedures (see *Common Practice Command Specification*). For example, Command 108 is used to select the command to be published and Command 109 turns publishing on and off (see Figure 8)

Command 109 is used to start Burst Mode publishing 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) advise the application that it 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 network has been configured, the Network Manager will return a response to the 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 Burst Mode is on.

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

Once initiated, the Field Device then published data indefinitely (potentially for years). The Field Device will continue to publish data until it 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.

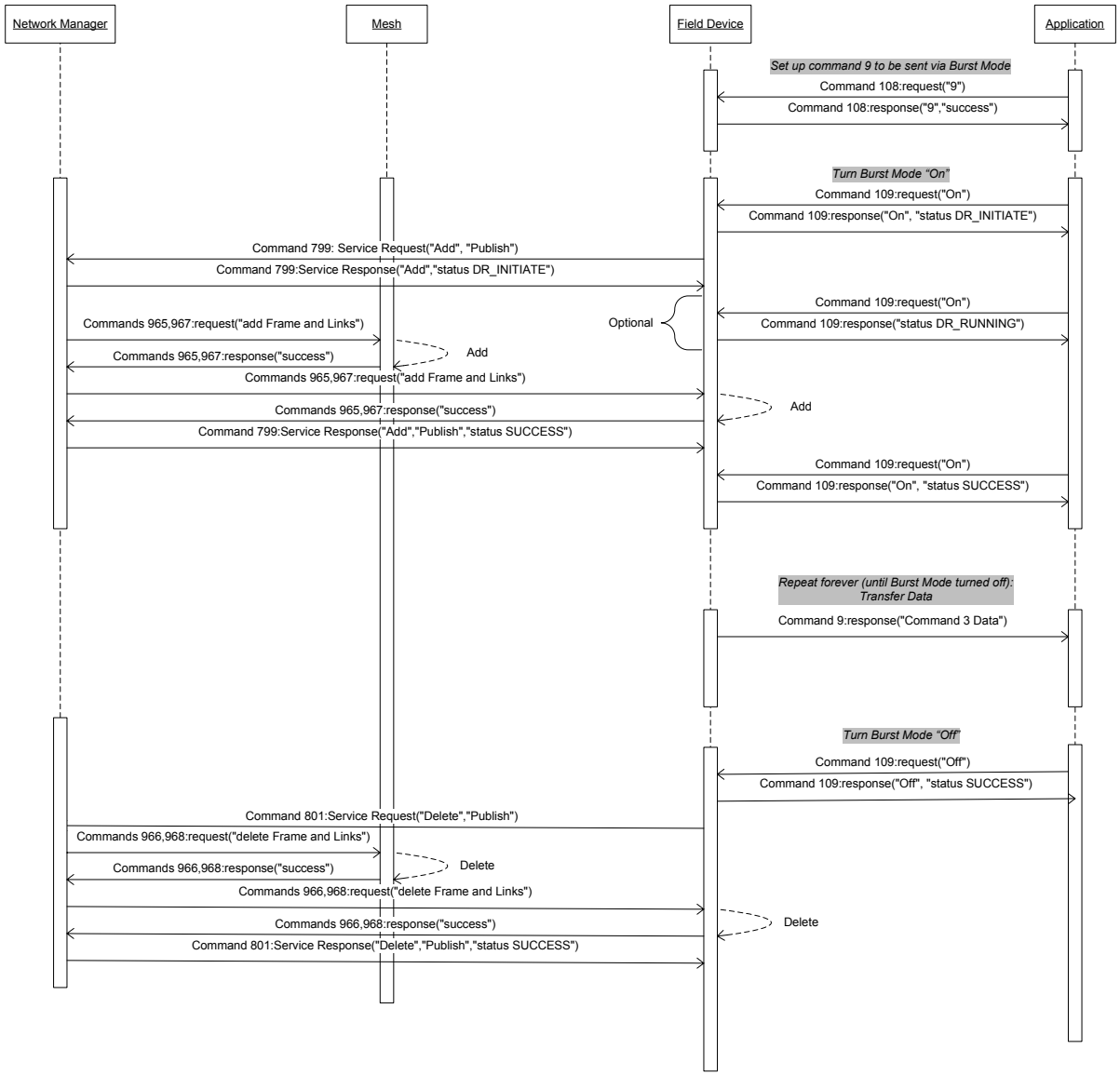


Figure 8. Supporting Burst Mode Data Transfer

6.3.3 Block Data Transfer

Block Data Transfer is used for segmented transfer of large data-sets between a Field Device and an application (see *Block Data Transfer Specification*). Using Command 111 the application opens a port for communication with a Field Device (see Figure 9). The Field Device returns a DR (see the *Command Response Code Specification*) to the application indicating that it working on the request. Then the Field Device issues an "Add Block Transfer" service request to the Network Manager. The Network Manager returns a DR to the Field Device and begins processing the request. The Network Manager determines the network resources to use and issues commands to set up Frames and Links. In most cases, the Network Manager will set up a high-speed superframe to handle Block Data transfers. Once network has be configured, the Network Manager will return a response to the Device indicating that the "Add Block Transfer" service request is complete. The Field Device will then complete the sequence by returning a response to the Application indicating that Command 110 is complete and the port is open.

As usual, the application may issue another Command 110 to the Field Device to monitor progress on the DR (see the Command Response Code Specification). If the command is incomplete, the Field Device will respond with the status "DR_RUNNING". If, due to network constraints, the Network Manager refuses the service request the block the field device must use the existing maintenance superframe to perform the block transfer.

Once the port is open, the Field Device and Application exchange Command 111 messages to move the data. Unlike operation using the Token-Passing Data-Link Layer, the principal data source (e.g, the field device) may issue multiple Command 111 packets without waiting peer acknowledgement. When the transmission is done, the Application or the Field Device will complete the block data transfer by issuing a Command 111 request to close the port. The Field Device will send a "Delete Block Transfer" 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.

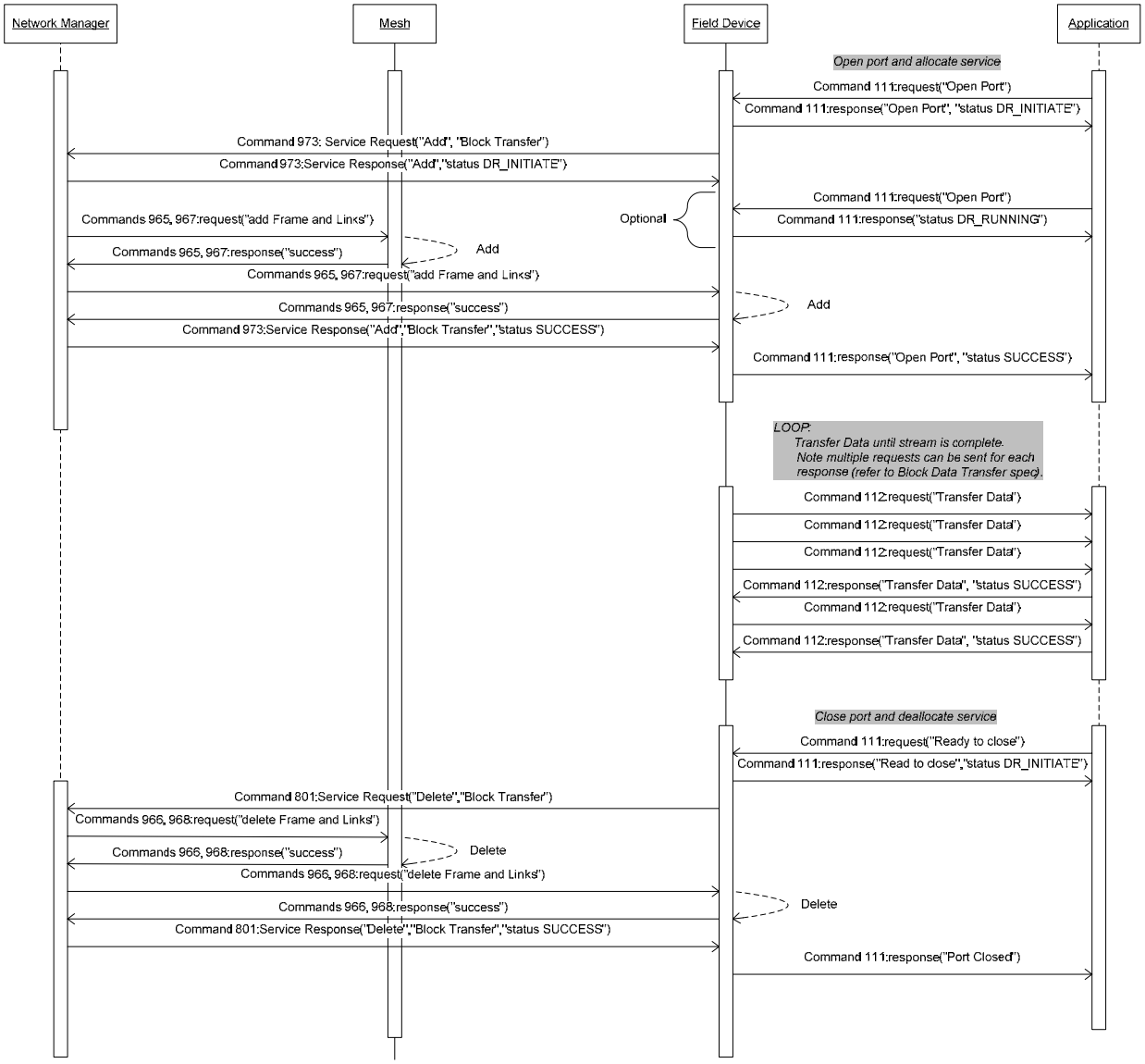


Figure 9. Supporting Block Data Transfer

When the initial request to start the Block Data Transfer is sent from the Application to the Field Device the device could respond in several ways. (e.g., if it is currently executing a Block Data Transfer it will respond with "Busy"). The Network Manager may also not be able to meet or fully satisfy the service request. If the Network Manager can respond in three ways:

- Success: the requested bandwidth was allocated
- Warning: a lower amount, than requested, of bandwidth is allocated
- Error: no extra bandwidth is allocated at all

In the worst case, i.e., when no extra bandwidth is allocated, the Field Device will have to use its "normal" data service to transfer the data. In this case, the Block Data Transfer could take considerable time to complete.

7. WIRELESSHART ADAPTERS

A WirelessHART Adapter (see Figure 10) connects to an existing HART compatible field device or to several multi-dropped field devices and enables communication to all of connected field devices via a WirelessHART Network. The adapter must contain both a (wired) Token-Passing and a (wireless) TDMA interface. In other words, and unless explicitly stated otherwise, a WirelessHART Adapter must meet all requirements for wired and WirelessHART communication. Furthermore, all WirelessHART Adapters must meet all the requirements found in this section.

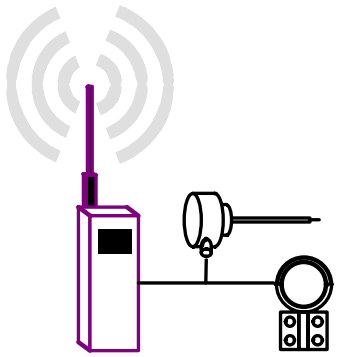


Figure 10. WirelessHART Adapter With Two Field Devices

The WirelessHART Adapter must support the publishing of process data and status on behalf of the connected field devices and allow complete access to the configuration and status of all the connected devices using Command 77 (see *Common Practice Command Specification*). In addition to supporting HART communication with the connected field devices, the adapter must have no adverse impact on analog signaling.

The Adapter must support Universal and Common Practice Commands and identifies itself (Manufacturer ID, Device Type Revision, Device ID, etc) in Identity Command responses. As with I/O Systems (see *Common Practice Command Specification*) the Adapter must set `Protocol_Bridge_Device` (bit 2) in the `Flags` byte of Identity Commands. In addition, the Adapter must support at least one Block Data Transfer connection (see *Block Data Transfer Specification*).

7.1 General Requirements

The Adapter builds on the I/O System requirements found in the *Common Practice Command Specification*. The procedures and commands found there provide support for sub-devices and a simple I/O System implementation. The commands provide the basic capabilities to identify and configure the I/O system, identify connected sub-devices and tunnel communications to them.

The Adapter must fully support the following Universal and Common Practice Commands found in Table 3. All Universal commands must be implemented.

Table 3. Required Common Practice Commands for Adapters

Common Practice

Cmd	Description
38	Reset Configuration Changed Flag
41	Perform Self Test
42	Perform Device Reset
48	Read Additional Status
72	Squawk
74	Read I/O System Capabilities
75	Poll Sub-Device
77	Send Command to Sub-Device
78	Read Aggregated Commands
84	Read Sub-Device Identity Summary
85	Read I/O Channel Statistics
86	Read Sub-Device Statistics
87	Write I/O System Master Mode
88	Write I/O System Retry Count
90	Read Real-Time Clock
101	Read Sub-device to Burst Message Map (see Subsection 6.3.2)

Cmd	Description
102	Map Sub-device to Burst Message
103	Write Burst Period
104	Write Burst Trigger
105	Read Burst Mode Configuration
106	Flush Delayed Response Buffers
107	Write Burst Device Variables
108	Write Burst Mode Command Number
109	Burst Mode Control
111	Transfer Service Control (on behalf of the devices)
112	Transfer Service
115	Read Event Notification Summary
116	Write Event Notification Bit Mask
117	Write Event Notification Timing
118	Event Notification Control
119	Acknowledge Event Notification

While not required, a Adapter may provide an analog interface to measure or control the loop current. When this is the case Commands 35, 40, 45, 46, 79 must be supported to allow access to and control of the loop current. The loop current must be mapped to a Device Variable (see *Command Summary Specification*).

Table 4. Recommended Common Practice Commands for Adapters

Common Practice

Cmd	Description
59	Write Number Of Response Preambles
71	Lock Device
72	Squawk

Cmd	Description
73	Find Device
76	Read Lock Device State
94	Read I/O System Client-Side Statistics

In addition to Universal and Common Practice Commands, the Field Device must support the standardized commands and procedures defined for WirelessHART devices.

If the sub-device is HART 7 (or later) then the adapter must be capable of synchronizing the sub-device to network time.

The Adapter must also provide the minimum capacity to support sub-devices as indicated in Table 5. For example, an Adapter may only indicate one I/O Card and one Channel in its Command 74

response. In addition to forwarding messages using Command 77, the Adapter must act as a proxy and publish commands on behalf of its sub-devices.

Table 5. 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 Burst (Data) Messages	5
Minimum Number of Burst (Event) Messages	2

7.2 Wired HART Interface

On the Token-Passing Data-Link, the Adapter is both a master and a slave. By being a master the Adapter can communicate with the connected HART devices, facilitate access to their configuration and acquire their process data for publishing to the Application. Functioning as a slave device allows the adapter to be configured using a legacy masters or host applications.

7.2.1 Master Operation

By default, the WirelessHART is a primary master and communicates continuously adhering to the requirements in the *Token-Passing Data-Link Layer Specification*.

If configured as secondary master the Adapter must be prepared to defer to a temporarily connected secondary master (e.g., a Handheld). In other words, if configured as a secondary master and another secondary master is detected; the Adapter must cease issuing master requests.

Note: Whenever possible, one of the connected devices must be configured into burst mode.

When a conflict with another secondary master occurs the Adapter must cease initiating transactions until no secondary master communication is detected for 4 times the Link Quiet Time (RT1). The Link Quiet Time for an Adapter shall be RT1(Secondary) plus RT2.

7.2.2 Slave Operation

In other words, Adapter will appear to be a multi-dropped device when another master is connected to its wired interface and polls for devices. By default, the Adapter should be configured for polling address 63.

All attributes and commands supported by the Adapter must be available via the wired interface. When requested, the Adapter answers Identity Commands normally thus allowing the DD-Enabled hosts to load the Adapter's DD.

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

Masters connected to the wired interface may access any device on the wired sub-network. However, connected masters do not have access to the wireless network.

Note: "Access restricted" shall be returned when the Adapter receives Command 75 via its wired interface.

7.3 WirelessHART Interface

The Adapter is also a WirelessHART device and must adhere to all WirelessHART specifications. Two additional key Adapter requirements include:

- Access to the connected wired devices via the I/O system commands; and
- Process data and status must be published by the Adapter on behalf of the connected sub-devices.

7.3.1 Timing requirements

All Adapters must be capable of routing messages on behalf of other Network Devices and its connected Sub-devices. Furthermore, communications must ensure latency across the mesh is minimized and unnecessary, redundant communications is minimized. To this end Adapters must:

- 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 Adapter in the first slot following the slot the PDU was received in.
- Be able to reply to all other commands addressed to the Adapter in the sixth slot following the slot the PDU was received in
- Where Delayed Responses are allowed by Commands to the Adapter, the DR_Initiate must not be generated until 75% of the Transport Layer maxReplyTime has elapsed.
- For commands addressed to the Adapter's Sub-devices, the Adapter must respond with the Sub-device response or a delayed response in 1 second.

7.3.2 Tunneling Commands to the Adapter's Sub-Devices

Command 77 allows Gateway access to the Adapter's sub-devices. The Gateway communicates on behalf of one or more Host Applications. The Host Applications connected to the Gateway do not (technically) communicate directly with the Adapter but rather tunnel messages through the Gateway or receive a cached response from the Gateway.

Note: For HART 5 devices connected to the Adapter, Long Tag shall be simulated using the "Message" attribute as read from Command 12.

Since response times on the wired sub-network are slower than WirelessHART, all Adapters must use the Delayed Response Mechanism (DRM) to indicate that the command was received and is being processed. As a bridging device, all commands may use the DRM and Adapters must support at least two Delayed Response buffers (one to the Gateway and one from connected Field Device).

Once the command for the sub-network is received, it is enqueued and transmitted on the wired connection in compliance with Token-Passing Data-Link requirements. If any of the following conditions occur, then the command must be sent retried.

- The field device does not respond.
- The field device received a communication error (Bit 7 set in the first response code byte).
- The I/O System encounters an error while receiving the response.

Otherwise, the response from the field sub-device is considered valid and will be returned to the Gateway. If the allowed number of retries is exceeded, DR_DEAD will be returned to the Gateway. A Busy or a DR response indicates communication is successful but the device is unable to complete the request. Consequently, the Adapter must continue to retry at least twice as many times as the specified "number of retries" used when a communication error is encountered.

7.3.3 Connected Device's Process Data and Status

The Adapter must publish (Burst Mode) data on behalf of the connected Field Devices (i.e., the sub-device). To accomplish this the Adapter contains a list of the commands to publish, the source device for the commands and the other required Burst Message attributes.

The Burst Mode is configured as required to meet the process or plant equipment requirements. This configuration is performed using standard HART procedures (see Common Practice Command Specification) with the following exceptions:

- All Burst Messages are configured in the Adapter (not in the sub-device)
- The source device for the Burst Message is specified using command 102.
- The Adapter is responsible for acquiring the data (as needed) from the sub-device.
- When the Burst Message Map indicates the data is from a sub-device, the command response from the sub-device is embedded in a Command 77 response and published to the Gateway.

Command 84 is used by Host Applications when configuring Burst messages. This command allows the identification of all potential sub-devices including the sub-device's: Manufacturer ID, Device Type, Device ID and Long Tag. Using Command 84, Host Applications present the list of sub-devices to the end-user. The end-user selects the source device, the command to be published and uses this information to configure the Adapter. Using Command 102, the source sub-device is specified.

The Adapter complies with the publishing requirements and generates the data on the specified schedule. When a command is published from a sub-device, a Command 77 response is generated. The Commands 77 response includes (in the data field) the address of the sub-device along with the balance of the command being published. This Command 77 response is indistinguishable from a Command 77 response originating from the sub-device itself.

8. WIRELESSHART GATEWAY

This section describes the WirelessHART Gateway. The Gateway is functionally divided into a Virtual Gateway and one or more Access Points. Multiple Access Point increases the throughput and the reliability of a WirelessHART Network. To simplify support for redundant Access Points, every Gateway has a fixed; well know 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 11.

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, should an Access Point fail, packets will flow to the remaining Access Points. Network traffic will be constrained but communication will still be successful.

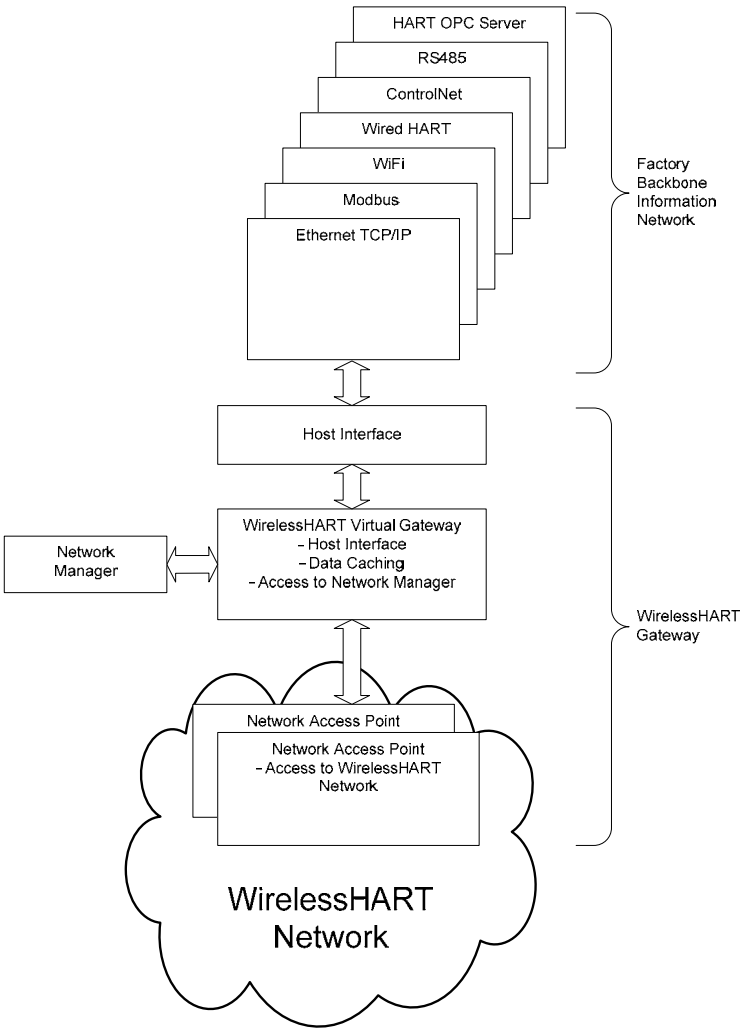


Figure 11. Gateway Scope

8.1 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:

- 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 Interfaccer connecting the Gateway to backbone networks (e.g., the plant automation network;
- An connection to the Network Manager;
- Buffering and local storage for burst mode, 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 also responsible for collecting and maintaining cached data and command responses from all devices in the network. These cached responses correspond to Burst 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.

The Gateway must natively to support Adapters (see Section 7) allowing transparent access to the Adapters sub-devices. The devices connected to the Adapter can be identified by polling the Adapter for its sub-devices (see the *Network Management Specification*). The Gateway uses Command 74 (see *Common Practice Command Specification*) to determine how many sub-devices may be connected to the Adapter. Then the Gateway issues Command 75 to walk through the legal combinations of polling addresses, I/O Card, and Channel identifiers until all the connected sub-devices are identified.

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 12) to eith a Host Interface or the Network Manager.

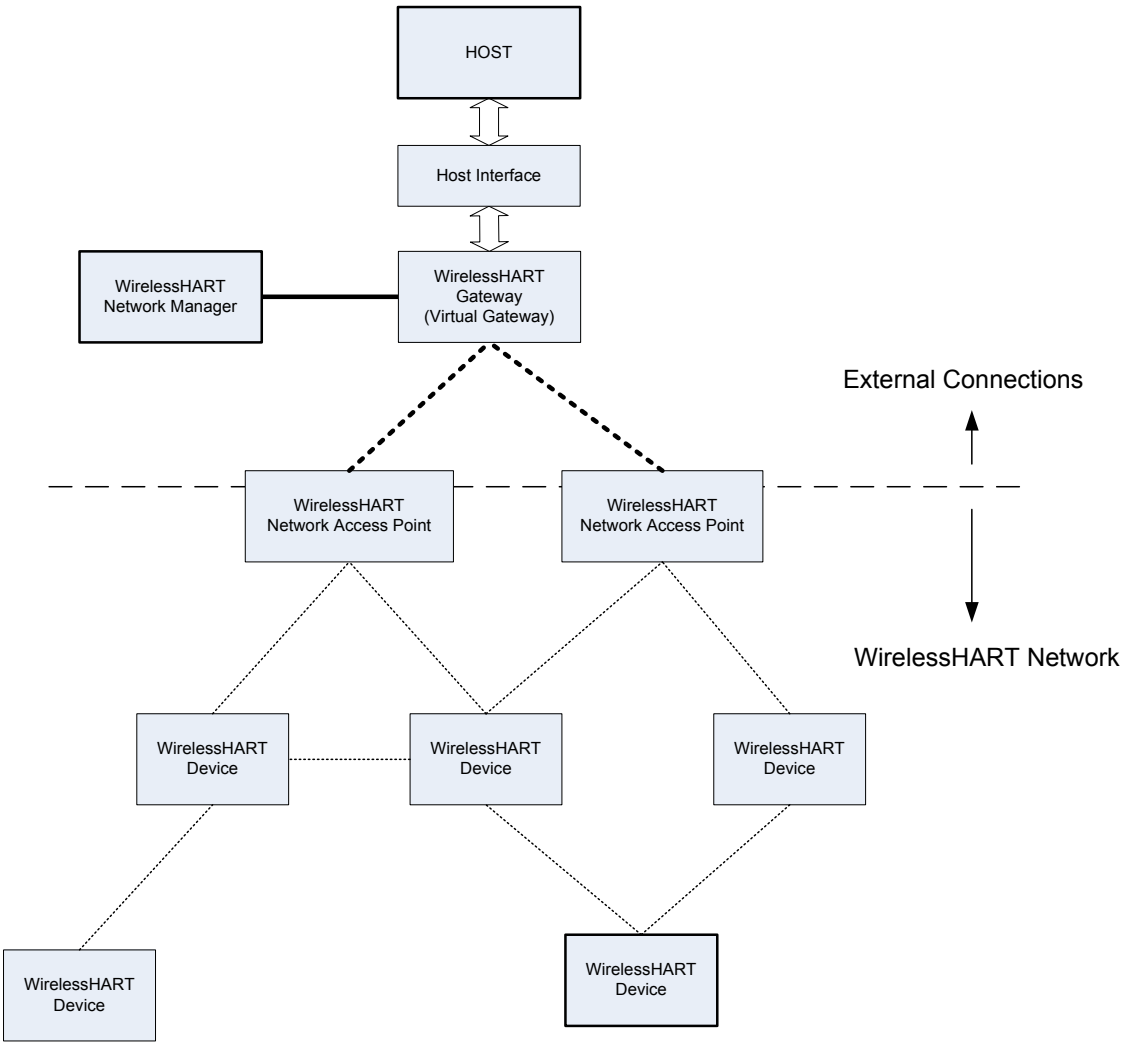


Figure 12. Virtual Gateway and Network Access Points in a WirelessHART Network

The WirelessHART Gateway must 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 must support Universal and Common Practice 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.

Example Gateway implementations are shown in Annex A.

8.2 Gateway Model

The WirelessHART Gateway has several distinct components as shown in Figure 13. 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.

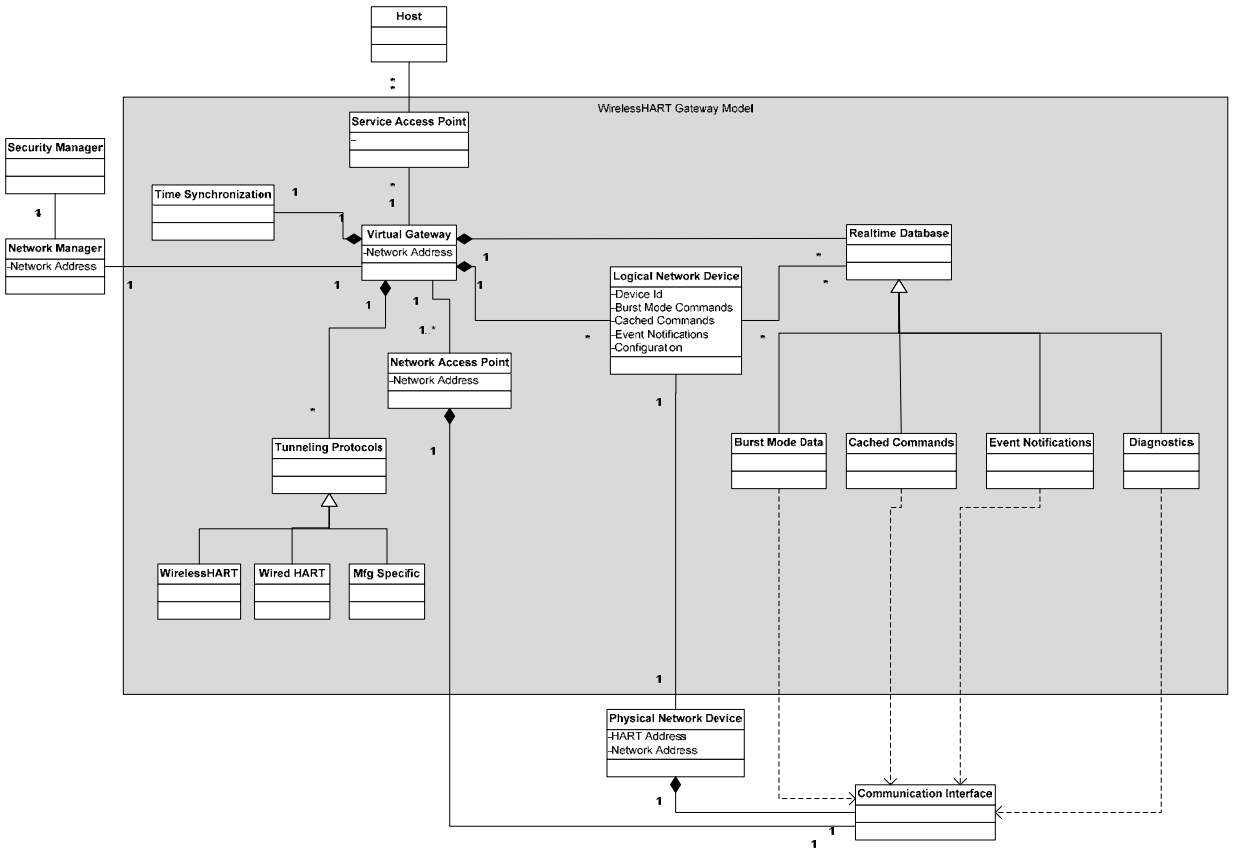


Figure 13. Gateway Model

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 sections.

8.2.1 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 must 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

  a. Described with DDL

  b. Supports HART DD

- 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. Burst Mode.

  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 burst mode 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 802.11 and 802.3.

8.2.2 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.

8.2.3 Service Access Point (SAP)

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. Burst Mode 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 SAP's the Virtual 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).

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

WirelessHART

A WirelessHART Gateway must be able to tunnel HART commands to/from any WirelessHART device. When a WirelessHART Adapter is in use in the network, the Gateway must also be able to tunnel HART commands to/from the wired HART device on the other side of the Adapter.

Vendor Specific Proprietary Protocols
A WirelessHART Gateway supporting open and vendor specific proprietary protocols must 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.

8.2.4 Tunneling Protocols

Gateways must 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 must be able to support HART and WirelessHART universal and common practice 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.

8.2.5 Host Interface

The Host interface is used to connect clients outside of the WirelessHART Network with the WirelessHART Network and devices in the WirelessHART Network. Host interfaces take on many forms – several common ones include:

- **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.

WirelessHART Gateways must be able to cache burst mode commands, several commonly used read and write commands, and diagnostics. To take advantage of this caching, the Gateway must also be able to act as a translator. As a translator, the Gateway peeks at requests from the Hosts and, if the response data is cached and is current, returns the cached response messages from its real-time database. For example, if a client on 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, 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.

8.2.6 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 Burst Mode Commands,

Event Notification Messages, and Cached Commands Responses that are currently stored there. The Logical Network Device is shown in Figure 14.

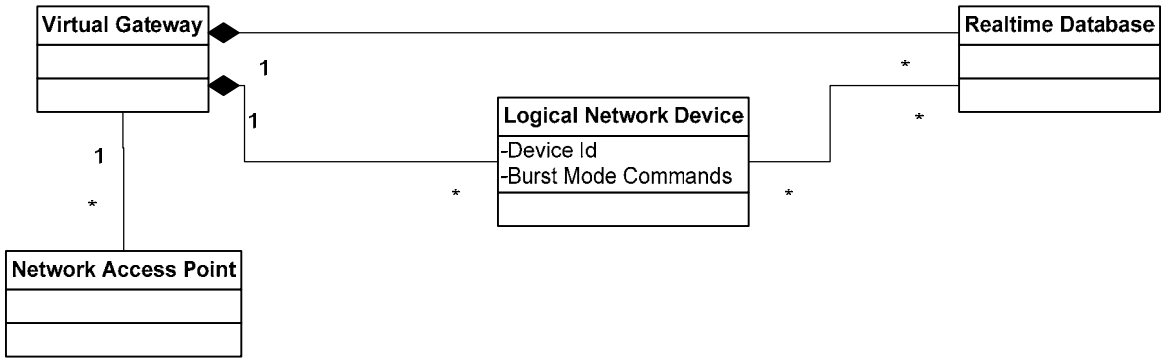


Figure 14. Logical Network Device

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.

8.2.7 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 15.

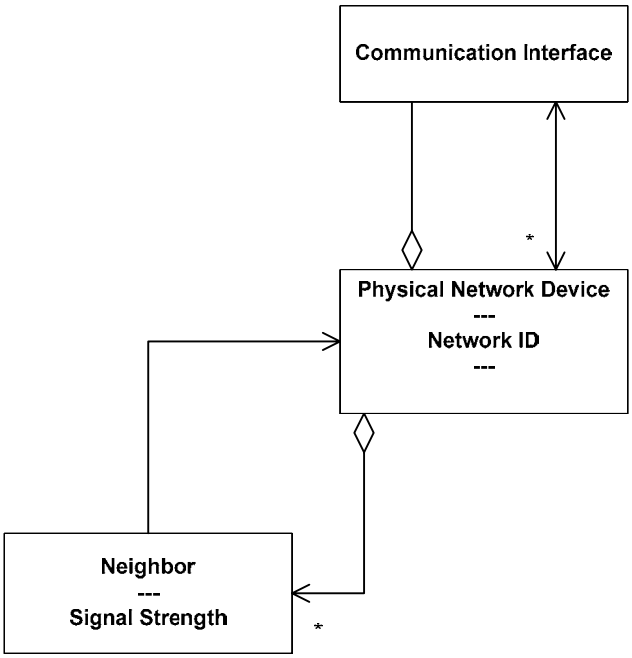


Figure 15. Physical Network Device

8.2.8 Connection between Gateway and Network Manager

This is an external connection

This specification does not specify the architecture of the Network Manager, Gateway, or Security Manager. For example, the Network Manager and Gateway may be physically combined or separate. In all designs, The Network Manager and the Gateway must establish and maintain a secure communication channel with each other. All communications with the WirelessHART Network pass through the Gateway. Consequently, the Gateway must 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.

8.2.9 Communication Interface

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

- Burst Messages
- Event Notification
- Diagnostics

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 must available to accommodate request/response traffic. The Gateway is responsible requesting this service, increasing and decreasing the bandwidth it requests as application demands vary.

Block Transfers
Block transfers must also be supported (see the Block Data Transfer Specification) both to Network Devices and through an Adapter to its sub-device. This supports transmission of large data sets with automatic segmentation and re-assembly the destination.

Burst Messages
The Burst Messages (see the *Common Practice Command Specification*) are used to send data on a periodic schedule or on an exception basis. A Burst Message consists of the response packet for the specified command. The burst mode rate is configurable, based on application needs.

For control applications, the burst mode 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.

Burst mode data must 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.

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.

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

8.2.10 Cached Response Messages

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.

Burst Mode Command Responses
The database caches all of the burst mode response messages.

Event Notification Command Responses
The database caches all of the event notification response messages.

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

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 must be purged if they exceed a 24 hr timeout.

8.3 Gateway Management

8.3.1 Addressing
Between the WirelessHART Gateway and the Host each instrument 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 HART devices, for example those connected through an Adapter, do not have a Nickname. In all cases the Gateway maintains a table of 5-byte HART addresses, Nicknames, and device types information. This table is used by the Gateway to translate addresses between Host (or Clients) and the network.

8.3.2 Retry Mechanisms
Result of the host application request to the Gateway may be classified:

- 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 a error in it that prevents it from being processed.

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

8.3.3 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 minutes, set an internal WARM_START flag. If the power outage time is greater than 15 minutes 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.

8.3.4 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.

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

Resetting the Gateway will 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 not 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.

8.3.6 Re-build Burst Mode 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 WirelessHART Adaptor, this means that all buffers for all devices connected to the Adapterwill also be reset. The WirelessHART Gateway will use command #0 to test the connection to any WirelessHART device and Command 75 to each Adapter Sub-Device.

8.3.7 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.

8.3.8 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.

Adapters and Sub-Devices

If the Network Device is an Adapter (i.e. a protocol bridge device) then the Gateway must determine the number of sub-devices connected to the Adapter and identify each of the sub-devices (see the *Common Practice Command Specification*). The Gateway determines which devices are Adapters by examining byte 8 (Flags), bit 2 (Protocol Bridge Device) of the Identity Command (Command 0).

The Gateway sends each sub-device a Command 0 and Command 20. The response messages for each sub-device are cached by the Gateway.

8.3.9 Device Configuration Change Status Notifications

Whenever the Gateway receives a Configuration Change Status, it must 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 an Adapter then the Gateway repeats the "Join Sequence for Adapter Sub-devices" connected to the Adapter.

8.4 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 then ChannelSearchTime (see the *Network Management Specification*). 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).

8.5 Gateway Change Notification Services

The Gateway can return change notification messages to the 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 burst mode 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 below:

Table 6. Required Common Practice Commands

Notification Type	Fastest Notification Rate (seconds)	Comments
BurstMode	0.250	The Gateway checks the cache on a device-by-device basis for burst mode data changes (e.g., cmd 9). If there are changes, a change notification is added to the notification list.
EventNotification	1	Every 1 second 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 seconds 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 burst mode message, when burst mode is enabled this rate will be increased up to the burst mode rate.
DeviceConfiguration	60	Every 60 seconds 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 burst, the notification should only be sent if no burst message has been sent.
NetworkTopology	60	Every 60 seconds 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 seconds 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 must supprt 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 burst mode 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 below:

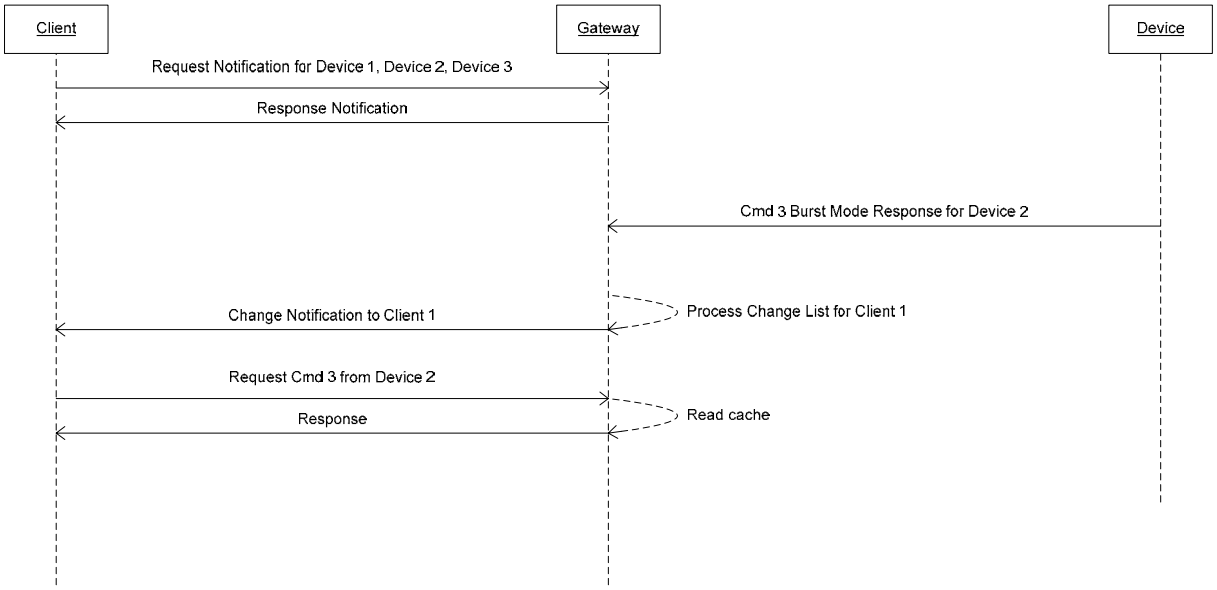


Figure 16. Managing Notification Services

8.6 HART Commands Interface

8.6.1 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 burst mode 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, burst mode, block mode transfers, delayed response messages, and high throughput services.
- Pass through commands requests and responses addressed to/from devices in the WirelessHART field 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 must retain it 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

8.6.2 Host to WirelessHART Command Request and Response

These are commands from a host and 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 must 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.

8.6.3 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 must 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.

8.6.4 Caching Burst Mode Command Response Messages

The dynamic data the WirelessHART Gateway records when a device sends burst mode data includes the data validity flags, byte count, response bytes, data bytes, total numbers of updates and the extended status bytes.

8.6.5 WirelessHART Gateway Status Error Flag Bits

The WirelessHART Gateway supports the following Status Error Flag Bits:

Table 7. WirelessHART Gateway Status Flags

Error	Command Descriptor
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 minutes; 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.

8.6.6 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 have the Gateway to run a self-test).

8.6.7 HART Capacities

The Gateway must also provide the minimum capacity to support devices as indicated in Table 8. There capacities are summarized below.

Table 8. Gateway Minimum Capacity Requirements

Parameter	Tiny (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	1000
Minimum Number of Packet Buffers	25	100	500
Minimum Number of Burst (Data) Messages	30	200	1000
Minimum Number of Cached Messages	75	400	2000
Minimum Number of Burst (Event) Messages	25	100	500
Total Number of Devices (Adapters + Field Devices)	10	50	250
Clients	4	8	32

The Gateway must cache response messages for the Adapter and for each sub-device.

8.6.8 HART Commands

The WirelessHART Gateway supports several Gateway specific commands, supports pass through messages, supports burst mode and block mode transfers, and caches several read and write commands. The commands are described in this section.

Gateway Supported Commands

The Gateway must support the Commands that are listed in the table below. The burst mode 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 must be supported and employed transparently by the Gateway. In other words, Adapter sub-devices must be accessed using the I/O system commands. For example, the Host applications requests must be automatically translated and embedded in Command 77 for tunneling through Adapters.

Table 9. Required Commands

Common Practice 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
74	Read I/O System Capabilities
75	Poll Sub-Device
77	Send Command to Sub-Device
78	Write Time of Day

Cmd	Description
84	Read Sub-Device Identity Summary
85	Read I/O Channel Statistics
86	Read Sub-Device Statistics
87	Write I/O System Master Mode
88	Write I/O System Retry Count
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
775	Write Network Tag
776	Read Network Tag
832	Read Network Information
833	Read Neighbor information
834	Read Network Topology Information
835	Read Burst Mode List

Cmd	Description
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

In addition, The Gateway must implement all of the Data Link Layer and the Network Layer Commands listed in the *Wireless Command Specification*.

Cached Response Messages

The Gateway must be able to cache the responses to the Commands listed below:

Table 10. 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
Burst Mode **	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 burst contains all 12 Trend values.
	Device specific	Any HART command (e.g. 132 for DVC5000 Valve Diagnostics)
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

Notes: * Field Devices must support commands 1, 2, 3, and 9 if Burst Mode is implemented.
** HART 7 devices must support a minimum of 3 Burst Messages. All Burst Mode responses from a device must be cached.

8.7 XML-based Interface (Optional)

8.7.1 XML-formatted Requests and Responses

The communication between a client and the WirelessHART Gateway can use well-known XML formatted messages as defined here in this section. These XML-based messages are exchanged using TCP on a Gateway specified TCP Port. The communication process entails the client sending an XML request message to the WirelessHART Gateway and receiving a corresponding XML response.

The client can request the following forms of communication traffic:

- Request/Response – the client sends a HART command request and receives back a HART command response.

- Change notification messages – the client registers for change notification messages on burst mode, event notification, device status changes, and HART Read/Write commands that are buffered by the Gateway. The WirelessHART Gateway periodically processes the change bits for each command on each device and sends change notification messages to each registered client.

- Block mode transfers – the client initiates a block mode transfer and the WirelessHART Gateway transfers the data to the device until the block mode transfer is complete.

- Delayed response – the client initiates an action in a device that will require a significant amount of time to complete (for example a method call). An initial response is generated with a delayed response code indicating the delayed response mechanism is being invoked. Subsequently, a second response will be returned when the requested action completes or times out. The client does not have to poll the Gateway for the final response, it is sent unsolicited.

    a. On the client/ server side of the Gateway, the interface supports unsolicited delayed responses to be sent without requiring the client to resend another request later on.

    b. On the Wireless side of the Gateway, the interface allows unsolicited delayed responses to be sent OTA without having to re-issue requests. .

    NOTE: The need for an efficient DRM is bi-directional. For example, a field device can issue a service request to the Network Manager that requires reconfiguration of the Gateway and several intermediate nodes. The Network Manager/Gateway need time to reconfigure the network. Without the ability to send an unsolicited response when it completes the action, the field device will have to poll the Network Manager/Gateway to see if the request has been honored.

The XML Responses from the WirelessHART Gateway to the XML Requests/Commands issued by the client are not coupled together in time. From the clients point-of-view sending XML-formatted commands to the Gateway is a non-blocking process; in the same manner XML responses are returned asynchronously. The client can issue several XML commands before any XML responses are returned from the Gateway (e.g. valid response, error, busy).

The WirelessHART Gateway is not required to issue an XML response in a timely way. Moreover there is no maximum time limit for which the WirelessHART Gateway must answer a command. However the WirelessHART Gateway is expected to respond to all client requests.

All XML messages are encoded using UTF-8 characters.

8.7.2 Service Session

A WirelessHART Gateway provides services to a client only after the Client and Gateway have established a Service Session. A WirelessHART Gateway makes known a specific IP Address and TCP Port number to which a Client will connect through to form a new Service Session.

A service session begins with a successful TCP socket connection to the Gateway. The service session ends when the socket connection is terminated. When the service session ends all active Leases between the Client and the Gateway also expire and all Network Resources associated with those Leases are released.

The Gateway is free to limit the number of concurrent Service Sessions. Although it is expected that a Service Session will exist and remain valid for a long period of time, the Gateway is free to terminate a Service Session without notice at any time.

8.7.3 Service Request

For the purposes of this document, a Service Request is a command sent from a Client to the Gateway for which the Gateway must perform some action. Service Requests are XML messages; more specifically XML with the form <*Request transactionId="123"* > ... </*Request*>. The XML 'Request' message format is defined in this document. TransactionId's are generated by the Client and the Gateway and must be unique within a Service Session.

8.7.4 Service Response

A Service Response is returned in response to a Service Request. Service Responses are XML messages of the form <*Response transactionId="123" status="success"*> ... </*Response*>. The XML 'Response' message format is defined in this document.

8.7.5 Identifier

An Identifier is used to uniquely itemize objects within a specific domain. The uniqueness of an Identifier is determined by the Identifier creator; which is also the owner or controller of objects in the domain. The value of an Identifier may have significant meaning to the creator. To all others, the meaning of the Identifier's value is opaque and is limited in usefulness to determining if two Identifiers are equal (the same item) or unequal (different items).

The form of an Identifier is a string of characters (UTF-8 as specified above), no less than 1 or more than 20 characters in length. Without restriction, all characters are legal to be used in an Identifier.

Some example Identifiers are:

```
1
10 39
Hello World!
Applesauce & Ham
03A3ED89
```

8.7.6 Network and Device Identifiers

A WirelessHART Gateway is a device in a WirelessHART network. Each WirelessHART Network is identified by a unique Network Identifier which is referred to as the **NetworkId**. The NetworkId is configured into the Gateway and used by the Network Manager to identify the network.

Each WirelessHART device is identified by its unique HART Address. The HART Address is used as the Device Identifier which is referred to as the **DeviceId**. **DeviceId**s are used by clients to identify a specific device to which a Service Request is applied.

8.7.7 Transaction Identifiers

Transactions are used to uniquely identify service request/response messages exchanged between Clients and Gateways. The initiator (either the Client or the Gateway) of a service request/response exchange is responsible for including a unique identifier in the request. The receiver of the request must include the same identifier in the response. These identifiers are called **TransactionId**s. Clients and Gateways are responsible for ensuring that all outstanding service requests have unique **TransactionId** values. **TransactionId** values must be unique within a specific Service Session.

There can be multiple outstanding transactions between a Client and a Gateway. An outstanding request is one that has been initiated by either a WirelessHART Gateway or a Client for which a response has not yet been received. Both the Client and the Gateway are responsible for maintaining a list of all outstanding transactions and performing periodic cleanups on these transactions by counting down the timer associated with the Lease Period and abandoning the transaction and returning a timed-out status to the request originator if the timer expires. Both the Client and the Gateway can abandon transactions. If the response message arrives after the transaction has been abandoned the response will be discarded.

TransactionIds are also used to detect duplicate Service Requests and Service Responses. There are two conditions when a Service Request might contain a duplicate **TransactionId** of an already received in-process Service Request. The first condition is when the Service Request is being purposefully sent again. Since this is a duplicate request, the Gateway or Client will send a normal response indicating that the response is still pending (note – this requires that all in-progress xml requests strings must be kept in order to do this check). The second condition in which a **TransactionId** will be duplicated is that a mistake has been made. In this case the newly received Service Request will not match the initial Service Request and an error response will be returned indicating that the **TransactionId** has been duplicated.

Transactions are used for service request/response communication traffic. For example, to receive change notification messages from a Gateway a Client must register for these changes. The registration process requires a service request/response message exchange with a unique **TransactionId**. Once the registration is complete change notification messages are sent from the Gateway to the Client. Each of the communication exchange types are summarized below.

Table 11. XML-Based Interface

Service Type	Initiated by Client	Initiated by Gateway	Description
Request/Response	Yes	Yes	The initiator of a service request/response transaction includes a unique TransactionId in the initial request message. The response message needs to indicate whether or not the transaction is complete (response could require several response messages). The Lease Period on transactions can be used to handle delayed response messages from devices (for example calibration request/response messages).
Configuration of Change Notification	Yes	No	Configuring change notification messages requires a service request/response exchange identified by a unique TransactionId. Once the change notification has been configured the Gateway publishes change notification messages for devices on the change list.
Change Notification Messages	No	Yes	The Gateway publishes change notification messages for devices and responses on its change list. Change Notification Messages contain the list of HART Command Responses that have changed since the last change notification messages were sent. The Change Notification Message includes the TransactionId that was used to set up change notification.
Burst Mode	Yes	No	Clients can request the Gateway to set up burst mode. When the Gateway allocates network resources for burst mode they are only maintained in existence for the duration of the lease. Two mechanisms will be used to track cache updates – a sequence number and a time stamp. The sequence number will be incremented each time a response message is cached. The time stamp when a response message was copied into the cache will be recorded. When the transaction is created the sequence number will be initialized to 0. Some response messages also include the originating time stamp from the device (e.g. Command 9).
Event Notification	Yes	No	Clients can request the Gateway to set up event notification. When the Gateway allocates network resources for event notification they are only in existence for the duration of the lease.

Service Type	Initiated by Client	Initiated by Gateway	Description
Block Mode	Yes	No	The initiator of a block mode request/response transaction includes a unique TransactionId in the initial request message. The response messages each include the initial TransactionId. The device must indicate when block mode transfer is complete. When block mode is set up a sequence number will be associated with the transaction and initialized to 0. Each time a response message is processed the sequence number will be incremented and the response message cached.
High Throughput Connection	Yes	No	The client requests the Gateway to allocate a high throughput connection (e.g. to support an asset management application). The high throughput setting instructs the Gateway to increase the network communication resources for a particular device. The high through connection is only in existence for the duration of the associated lease.
Configuration	Yes	No	Support Gateway configuration requests.

As indicated in the table above, under some circumstances the Gateway initiates service request/response message exchanges with clients. The Gateway can initiate service request/response messages when it needs to change a specific service that has already been configured between the client and the Gateway.

8.7.8 Leases

The Gateway acts as a server to client applications allowing them access to assets on the WirelessHART network. The Gateway provides connection-oriented services to the client applications. When the XML interface is supported, connections are managed using Leases.

Lease Overview

All WirelessHART Network resources allocated by the Gateway to support Client Application's service requests and service responses are managed using leases. When the lease expires, the network resources are released back to the system. For example, when a client application allocates a high throughput connection the Gateway will make a request to the Network Manager that allocates the communication resources (Links and Frames). When the lease expires, the Gateway will make another request to the Network Manager to remove those Links and Frames.

Leases are set up through transactions. Leases are uniquely identified by their leaseId. The following example illustrates how leases are requested:

```
<Request transactionId="unique id 1">
  <LeaseRequest leaseId="123" leasePeriod="600000" deviceId="1">
  </LeaseRequest>
</Request>
```

Leases are used to manage network resources. The amount of time a lease is kept around is called a Lease Period. A Lease can be expressly canceled before the Lease Period expires. The Client and the Gateway are called the Lease Holder and the Lease Grantor. A Lease Holder has the resource allocated to their use. The Lease Granter owns the resource and leases its use to the Lease Holder. The Lease Grantor, i.e. Gateway, must request network resources from the Network Manager and target devices (for example, a client may request the gateway to set up burst mode on a specific command on a specific device – the request could fail if either the Device or the Network Manager are unable to comply with the request).

The Lease Period is measured in seconds and its value is zero or more. The Lease Period is always measured from the present time into the future. The period is always relative to the Lease Granter's point of view. The Lease Period starts when the Lease Granter issues the Lease, not when the Lease Holder receives the message that the Lease has been granted.

When a Lease for a resource is requested, the potential Lease Holder specifies the desired Lease Period for which it would like to have access to the resource; this however is merely a request. The Lease Granter is free to reduce the requested Lease Period before granting the Lease. The Lease Granter never increases the requested Lease Period. The return message sent to the potential Lease Holder indicates that the Lease has been granted or denied, contains the definitive value for the Lease Period (possibly reduced from the request) for which the lease will be valid. A return message with a Lease Period value of zero seconds indicates that the Lease has been denied and the resource has not been allocated. Any other positive value indicates that the Lease has been granted and the resource has been allocated to the Lease Holder.

A Lease for a resource can be renewed by the Lease Holder before the Lease Period expires. This maintains the allocation of the resource for a longer period of time. A Lease renewal is accomplished in the same way as the original request. Before the Lease expires, the Lease Holder sends another Lease request for the same resource to the Lease Grantor (note – any lease request/response supersedes previous lease agreement.). As long as these subsequent Lease requests occur before the current Lease Period expires, the Lease can be extended indefinitely as long as the Lease Grantor is willing to concur.

Leases can be prematurely canceled by either the Lease Holder or Lease Granter by sending the other party a Lease request with a Lease Period value of zero seconds.

When a Lease expires, no notices or messages are sent or exchanged between either the Lease Holder or Lease Grantor.

Lease Identifier

All active Leases are identified by unique identifiers. These identifiers are called **LeaseId**s. An active Lease is one that has been successfully negotiated with the Gateway and which has not yet expired or is a Lease in the process of being negotiated. For example, there would be a specific LeaseId for HighThroughput communication to device1, and a separate LeaseId for HighThroughput communication to device2.

Leases are associated with network resources allocated to address the increased communication resources for a specific Device (for example a Superframe and additional Links are allocated to support a block mode transfer). A specific **LeaseId** must be used with the DeviceId for which it was allocated, i.e., the lease cannot be used in context with a different device. When a **LeaseId** is used in

association with the wrong device the Gateway will return an error response to the Client indicating that the **LeaseId** has been used incorrectly.

Lease Request

This diagram illustrates the interaction between the Client and the Gateway when the Client requests the Gateway for a high throughput connection between the Gateway and a Device. Examples of lease types include:

Table 12. Typical Lease Types

	Lease Type	Comments
	HighThroughput	A high throughput lease type tells the gateway that communication resources need to be allocated to support a large number of packets – for example to support an asset management application or a block mode transfer.
	BurstMode	Clients can request the gateway to configure burst mode. A burst mode lease type tells the gateway that communication resources need to be allocated to support period traffic. The Network Manager will determine whether a dedicated superframe or additional links are sufficient to satisfy the request.
	EventNotification	Clients can request the gateway to set up event notification. A event notification lease type tells the gateway that communication resources need to be allocated to support event traffic. The Network Manager will determine additional links are required to support the request.

In the following example, the Gateway grants the lease but shortens the lease period to a third of the requested length. After a period of time but before the Lease is due to expire, the Client attempts to renew the Lease for an additional length of time. In this case, the Gateway denies the renewal by shortening the granted lease period to zero seconds.

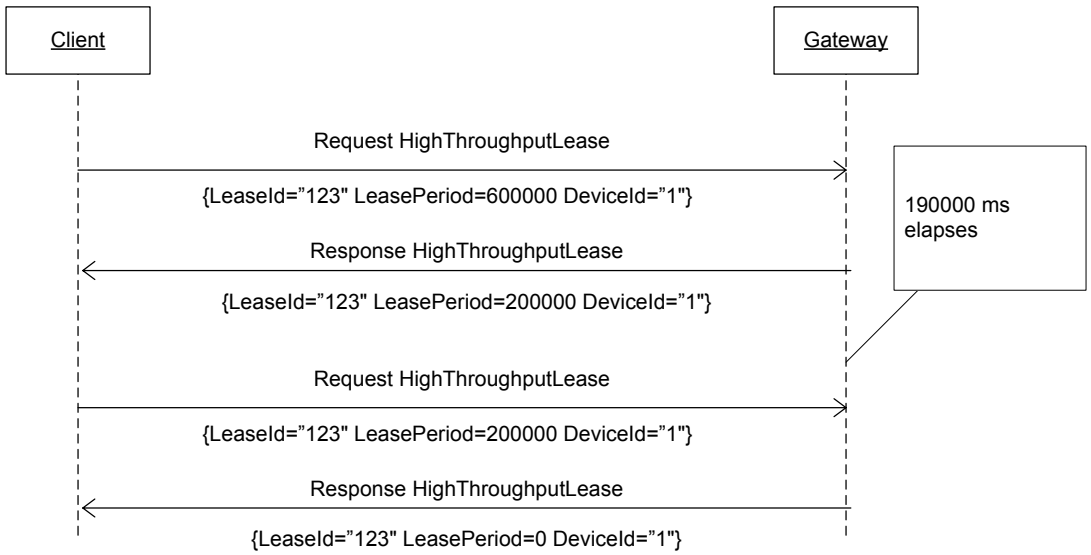


Figure 17. Request High Throughput Lease

The XML-based service request and response for this interaction are summarized below:

```
<Request transactionId="unique id 1">
  <LeaseRequest leaseId="123" leasePeriod="6000000" deviceId="1" leaseType="HighThroughputLease">
  </LeaseRequest>
</Request>

<Response transactionId="unique id 1" status="success">
  <LeaseRequest leaseId="123" leasePeriod="6000000" deviceId="1" leaseType="HighThroughputLease">
</Response>

<Request transactionId="unique id 2">
  <LeaseRequest leaseId="123" leasePeriod="2000000" deviceId="1" leaseType="HighThroughputLease">
</Request>

<Response transactionId="unique id 2" status="success">
  <LeaseRequest leaseId="123" leasePeriod="0" deviceId="1" leaseType="HighThroughputLease">
</Response>
```

8.7.9 Gateway Discovery

In a process plant there can be many Gateways. Clients need a way to discover the list of Gateways and from that list select the Gateway that they are interested in. To address this need Gateways are uniquely identified using GUID.

A WirelessHART Gateway makes itself known to potential clients that are seeking its services by utilizing a UDP multicast process. Using a well-known multicast group, the WirelessHART Gateway listens for discovery queries from potential clients. When a multicast message is received from a potential client, the content of the message is inspected for a well-known discovery query GUID. The Gateway shall unicast a UDP response message to the client only when the received multicast message contains the recognizable discovery query GUID. The unicast UDP response message contains the discovery GUID, IP Address and TCP Port number, in this order. The IP Address and TCP Port number specifies where the connection shall be made by the Client in order to establish a Service Session with the Gateway.

The multicast group and port used to discover WirelessHART Gateways are configurable parameters in the Gateway. The discovery query GUID is also configurable in the Gateway.

The entire data content of the multicast UDP packet consists of the Discovery Query GUID. The GUID is a 16 byte (128 bit) binary value (more fully explained in http://en.wikipedia.org/wiki/GUID). The value of the GUID's five parts is sent in binary form in the IP's network order (bytes ordered from left to right). For example, the 5 part Discovery Query GUID could be:

```
{4C0F38AC-48AE-4935-B689-8F21F85FC030}
```

This Query GUID would have the following 16 byte binary form in IP network order:

```
AC 38 0F 4C AE 48 35 49 89 B6 8F 21 F8 5F C0 30
```

The Discovery Response unicast UDP packet sent from the Gateway specifies the Gateway's IP address followed by the Port number; in IP network order (bytes ordered from left to right). The IP address is four bytes long and is conformant to IP Version 4 format. The Port Number is a 2 byte

unsigned integer value. The entire data payload of a Discovery Response contains 22 bytes in total (16 byte GUID followed by 4 byte IP address followed by 2 byte port number).

The following diagram illustrates the exchange of Discovery UDP messages between a Client and a Gateway during the Gateway discovery process.

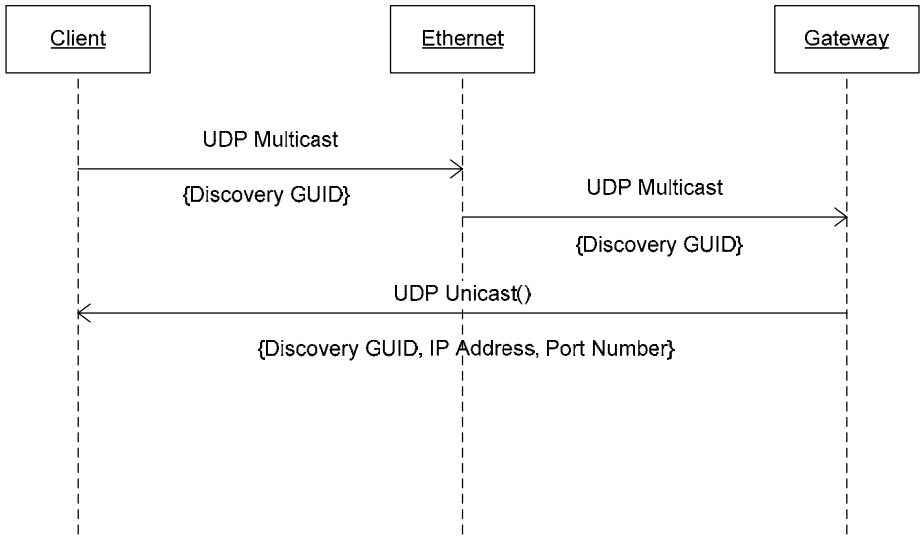


Figure 18. Gateway Discovery

8.7.10 Request / Response Messages Handled by Gateway

This section defines the Service Request/Response message types that are exchanged between a Client and a Gateway. The following simple network is used in most of the examples.

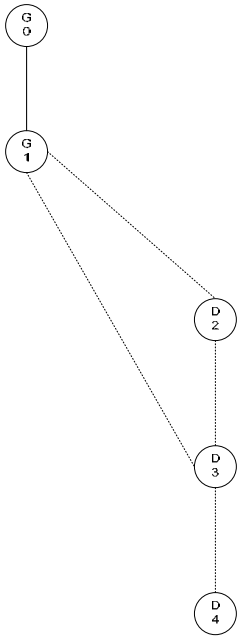


Figure 19. Example Network

List of Devices

The following diagram illustrates a Request / Response sequence requesting the list of known WirelessHART Devices connected either directly or indirectly to the Gateway.

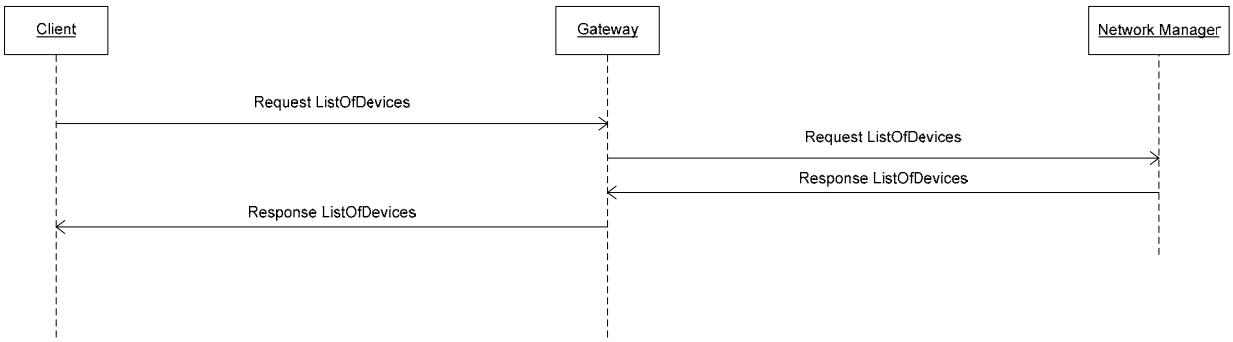


Figure 20. List of Devices

The service request and service response are summarized below:

```
<?xml version="1.0" encoding="utf-8"?>
<Request transactionId="01245">
  <ListOfDevices/>
</Request>

<?xml version="1.0" encoding="utf-8"?>
<Response transactionId="01245" status="success">
< ListOfDevices>
  <NetworkDevice address="0" type="VirtualGateway" id="0xC9C72D2F938E5800">
    <Neighbor address="1" signalQuality="100" />
  </NetworkDevice>
  <NetworkDevice address="1" type="Gateway" id="0x52BD9854A57B3000">
    <Neighbor address="0" signalQuality="100" />
    <Neighbor address="2" signalQuality="85" />
    <Neighbor address="3" signalQuality="30" />
  </NetworkDevice>
  <NetworkDevice address="2" type="Device" id="0xCDA62C139B4C5800">
    <Neighbor address="1" signalQuality="85" />
    <Neighbor address="3" signalQuality="65" />
  </NetworkDevice>
  <NetworkDevice address="3" type="Device" id="0x5F753BCABEEA7800">
    <Neighbor address="1" signalQuality="30" />
    <Neighbor address="2" signalQuality="65" />
    <Neighbor address="4" signalQuality="61" />
  </NetworkDevice>
  <NetworkDevice address="4" type="Device" id="0x8436EA85086DD800">
    <Neighbor address="3" signalQuality="61" />
  </NetworkDevice>
</ ListOfDevices>
</Response>
```

Note: This request must be directed to the Network Manager as only the Network Manager has all of this information.

Network Topology

The following diagram illustrates a Request / Response sequence requesting the list of known WirelessHART Devices and their hierarchal relationships.

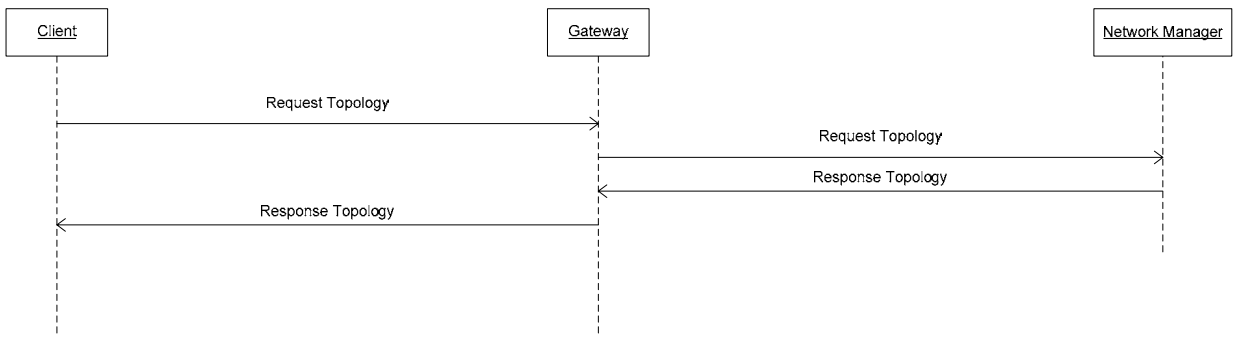


Figure 21. Network Topology

The service request and response are summarized below:

```
<?xml version="1.0" encoding="utf-8"?>
<Request transactionId="01245">
  <Topology/>
</Request>


?xml version="1.0" encoding="utf-8"?>
<Response transactionId="01245" status="success">
  <Topology>
    <NetworkDevice address="0" type="VirtualGateway" id="0xC9C72D2F938E5800">
      <Neighbor address="1" signalQuality="100"/>
      <!--Downstream graph-->
      <GraphRoute graphId="2">
        <GraphNode address="1"/>
      </GraphRoute>
      <!--Graph to node 1-->
      <GraphRoute graphId="11">
        <GraphNode address="1"/>
      </GraphRoute>
      <!--Graph to node 2-->
      <GraphRoute graphId="12">
        <GraphNode address="1"/>
      </GraphRoute>
      <!--Graph to node 3-->
      <GraphRoute graphId="13">
        <GraphNode address="1"/>
      </GraphRoute>
      <!--Graph to node 4-->
      <GraphRoute graphId="14">
        <GraphNode address="1"/>
      </GraphRoute>
    </NetworkDevice>
    <NetworkDevice address="1" type="Gateway" id="0x52BD9854A57B3000">
      <Neighbor address="0" signalQuality="100"/>
      <Neighbor address="2" signalQuality="85"/>
      <Neighbor address="3" signalQuality="30"/>
      <!--Upstream graph-->
      <GraphRoute graphId="1">
        <GraphNode address="0"/>
      </GraphRoute>
      <!--Downstream graph-->
      <GraphRoute graphId="2">
        <GraphNode address="2"/>
      </GraphRoute>
      <!--Graph to node 2-->
      <GraphRoute graphId="12">
        <GraphNode address="2"/>
      </GraphRoute>
      <!--Graph to node 3-->
      <GraphRoute graphId="13">
        <GraphNode address="2"/>
      </GraphRoute>
      <!--Graph to node 4-->
```

```
    <GraphRoute graphId="14">
      <GraphNode address="2"/>
    </GraphRoute>
  </NetworkDevice>
  <NetworkDevice address="2" type="Device" id="0xCDA62C139B4C5800">
    <Neighbor address="1" signalQuality="85"/>
    <Neighbor address="3" signalQuality="65"/>
    <!--Upstream graph-->
    <GraphRoute graphId="1">
      <GraphNode address="1"/>
    </GraphRoute>
    <!--Downstream graph-->
    <GraphRoute graphId="2">
      <GraphNode address="3"/>
    </GraphRoute>
    <!--Graph to node 3-->
    <GraphRoute graphId="13">
      <GraphNode address="3"/>
    </GraphRoute>
    <!--Graph to node 4-->
    <GraphRoute graphId="14">
      <GraphNode address="3"/>
    </GraphRoute>
  </NetworkDevice>
  <NetworkDevice address="3" type="Device" id="0x5F753BCABEEA7800">
    <Neighbor address="1" signalQuality="30"/>
    <Neighbor address="2" signalQuality="65"/>
    <Neighbor address="4" signalQuality="61"/>
    <!--Upstream graph-->
    <GraphRoute graphId="1">
      <GraphNode address="2"/>
    </GraphRoute>
    <!--Downstream graph-->
    <GraphRoute graphId="2">
      <GraphNode address="4"/>
    </GraphRoute>
    <!--Graph to node 4-->
    <GraphRoute graphId="14">
      <GraphNode address="4"/>
    </GraphRoute>
  </NetworkDevice>
  <NetworkDevice address="4" type="Device" id="0x8436EA85086DD800">
    <Neighbor address="3" signalQuality="61"/>
    <!--Upstream graph-->
    <GraphRoute graphId="1">
      <GraphNode address="3"/>
    </GraphRoute>
  </NetworkDevice>
 </Topology>
</Response>
```

This request must be directed to the Network Manager – only the Network Manager has all of this information.

Network Schedule

The following diagram illustrates a Request / Response sequence requesting the network schedule for a particular device or the whole network. If the schedule for a specific device is required include the device(s) in the Request Message.

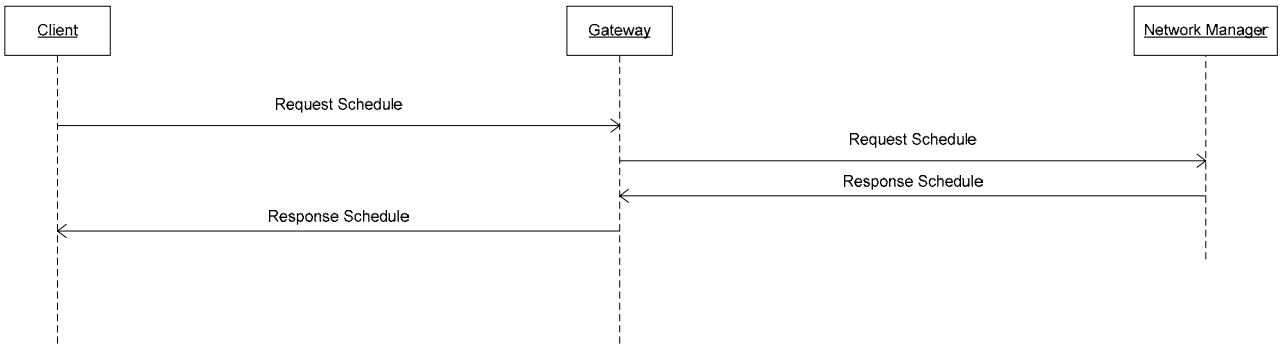


Figure 22. Network Schedule

The service request and response are summarized below:

```
<?xml version="1.0" encoding="utf-8"?>
<Request transactionId="01245">
  <Schedule>
    <DeviceAddress> 4 </DeviceAddress>
  </Schedule>
</Request>


<?xml version="1.0" encoding="utf-8"?>
<Response transactionId="01245" status="success">
  <Schedule>
     <ChannelMap>
        <Channel channelNumber="0"> enabled </Channel>
        <Channel channelNumber="1"> enabled </Channel>
        <Channel channelNumber="2"> enabled </Channel>
        <Channel channelNumber="3"> enabled </Channel>
        <Channel channelNumber="4"> enabled </Channel>
        <Channel channelNumber="5"> enabled </Channel>
        <Channel channelNumber="6"> enabled </Channel>
        <Channel channelNumber="7"> enabled </Channel>
        <Channel channelNumber="8"> enabled </Channel>
        <Channel channelNumber="9"> enabled </Channel>
        <Channel channelNumber="10"> enabled </Channel>
        <Channel channelNumber="11"> enabled </Channel>
        <Channel channelNumber="12"> enabled </Channel>
        <Channel channelNumber="13"> enabled </Channel>
        <Channel channelNumber="14"> enabled </Channel>
        <Channel channelNumber="15"> enabled </Channel>
    </ChannelMap>
    <NetworkDevice address="4" type="Device" id="0x8436EA85086DD800">
```

```
<!--Frames-->
<Frames>
 <Superframe Id="0" NumTimeSlots="6000" ActiveFlag="Active">
  <Links>
   <Link NeighborId="*">
    <TimeSlot>0</TimeSlot>
    <ChOffset>3</ChOffset>
    <LinkOptions> Receive </LinkOptions>
    <LinkType> Advertise </LinkType>
   </Link>
   <Link NeighborId="1">
    <TimeSlot>7</TimeSlot>
    <ChOffset>3</ChOffset>
    <LinkOptions> Transmit </LinkOptions>
    <LinkType> Normal </LinkType>
   </Link>
   <Link NeighborId="1">
    <TimeSlot>15</TimeSlot>
    <ChOffset>1</ChOffset>
    <LinkOptions> Receive </LinkOptions>
    <LinkType> Normal </LinkType>
   </Link>
   <Link NeighborId="1">
    <TimeSlot>16</TimeSlot>
    <ChOffset>3</ChOffset>
    <LinkOptions> Transmit </LinkOptions>
    <LinkType> Normal </LinkType>
   </Link>
   <Link NeighborId="1">
    <TimeSlot>21</TimeSlot>
    <ChOffset>1</ChOffset>
    <LinkOptions> Receive </LinkOptions>
    <LinkType> Normal </LinkType>
   </Link>
  </Links>
 </Superframe>
 <Superframe Id="4" NumTimeSlots="400" ActiveFlag="Active">
  <Links>
   <Link NeighborId="*">
    <TimeSlot>0</TimeSlot>
    <ChOffset>3</ChOffset>
    <LinkOptions> Receive </LinkOptions>
    <LinkType> Advertise </LinkType>
   </Link>
   <Link NeighborId="1">
    <TimeSlot>2</TimeSlot>
    <ChOffset>0</ChOffset>
    <LinkOptions> Transmit </LinkOptions>
    <LinkType> Normal </LinkType>
   </Link>
   <Link NeighborId="1">
    <TimeSlot>3</TimeSlot>
    <ChOffset>0</ChOffset>
    <LinkOptions> Transmit </LinkOptions>
    <LinkType> Normal </LinkType>
   </Link>
   <Link NeighborId="2">
    <TimeSlot>4</TimeSlot>
```

```
          <ChOffset>3</ChOffset>
          <LinkOptions> Transmit </LinkOptions>
          <LinkType> Normal </LinkType>
         </Link>
        </Links>
       </Superframe>
     </Frames>
   </NetworkDevice>
  </Schedule>
</Response>
```

Note - this request must be directed to the Network Manager – only the Network Manager has all of this information.

Device Health Report

The Device Health Report can be requested for a single device, a collection of devices, or all devices. If no devices are specified, the default is all devices. The response for all devices is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<Request transactionId="01245">
  <DeviceHealthReport/>
</Request>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="01245" status="success">
  <DeviceHealthReport>
    <NetworkDevice address="1" type="Gateway" id="0x3CC0CDD079819C00">
      <Packets Generated="200" Recieved="200"/>
      <MIC MacMicFailures="1" NetworkMicFailures="10"/>
    </NetworkDevice>
    <NetworkDevice address="2" type="Device" id="0x26C76E224D8EDC00">
      <Packets Generated="200" Recieved="200"/>
      <MIC MacMicFailures="1" NetworkMicFailures="10"/>
    </NetworkDevice>
    <NetworkDevice address="3" type="Device" id="0xC654B9118CA97000">
      <Packets Generated="200" Recieved="200"/>
      <MIC MacMicFailures="1" NetworkMicFailures="10"/>
    </NetworkDevice>
    <NetworkDevice address="4" type="Device" id="0x175C81142EB90200">
      <Packets Generated="200" Recieved="200"/>
      <MIC MacMicFailures="1" NetworkMicFailures="10"/>
    </NetworkDevice>
  </DeviceHealthReport>
</Response>
```

This request must be directed to the Network Manager – only the Network Manager has all of this information.

Neighbor Health Report

The Neighbor Health Report can be requested for a single device, a collection of devices, or all devices. If no devices are specified then the default is all devices. The response for devices 2 is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="01245">
  <NeighborHealthReport>
    <DeviceAddress> 2 </DeviceAddress>
  </NeighborHealthReport>
</Request>

<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="01245" status="success">
  <NeighborHealthReport>
    <DeviceAddress address="1" type="Gateway" id="0x3CC0CDD079819C00">
      <Diagnostics>
        <pathFailure>1</pathFailure>
        <receiveSignalLevel>7</receiveSignalLevel>
        <failedTransmitsToThisNeighbor>1</failedTransmitsToThisNeighbor>
        <packetsTransmittedToThisNeighbor>100</packetsTransmittedToThisNeighbor>
        <packetsRecievedFromThisNeighbor>200</packetsRecievedFromThisNeighbor>
      </Diagnostics>
    </DeviceAddress>
    <DeviceAddress address="3" type="Device" id="0xC654B9118CA97000">
      <Diagnostics>
        <pathFailure>1</pathFailure>
        <receiveSignalLevel>7</receiveSignalLevel>
        <failedTransmitsToThisNeighbor>1</failedTransmitsToThisNeighbor>
        <packetsTransmittedToThisNeighbor>100</packetsTransmittedToThisNeighbor>
        <packetsRecievedFromThisNeighbor>200</packetsRecievedFromThisNeighbor>
      </Diagnostics>
    </DeviceAddress>
  </NeighborHealthReport>
</Response>
```

This request must be directed to the Network Manager – only the Network Manager has all of this information.

WirelessHART Network Health Report

The WirelessHART Network Health Report can be requested for a single device, a collection of devices, or all devices. If no devices are specified then the default is all devices. The response for devices 1 and 2 are shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<Request transactionId="01245">
  < WirelessNetworkReport >
    <DeviceAddress> 1 </DeviceAddress>
    <DeviceAddress> 2 </DeviceAddress>
  </ WirelessNetworkReport >
</Request>

<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="01245" status="success">
  <WirelessNetworkReport>
    <Network networkId="102" type="Mesh">
      <StartDate>1/1/2007</StartDate>
      <CurrentDate>3/30/2007</CurrentDate>
      <PacketsSent>115900</PacketsSent>
      <PacketsLost>10</PacketsLost>
      <DataReliability>99.998</DataReliability>
      <Latency>100</Latency>
      <PathReliability>82.39</PathReliability>
      <NumberOfDevices>4</NumberOfDevices>
      <Joins>8</Joins>
    </Network>
    <NetworkDevice address="1" type="Gateway" id="0x3CC0CDD079819C00">
      <StartDate>1/1/2007</StartDate>
      <CurrentDate>3/30/2007</CurrentDate>
      <PacketsSent>100000</PacketsSent>
      <PacketsLost>1</PacketsLost>
      <DataReliability>99.998</DataReliability>
      <Latency>100</Latency>
      <PathReliability>99.5</PathReliability>
      <NumberOfDevices>3</NumberOfDevices>
      <Joins>5</Joins>
    </NetworkDevice>
    <NetworkDevice address="2" type="Device" id="0x26C76E224D8EDC00">
      <StartDate>1/5/2007</StartDate>
      <CurrentDate>3/30/2007</CurrentDate>
      <PacketsSent>10000</PacketsSent>
      <PacketsLost>1</PacketsLost>
      <DataReliability>99.998</DataReliability>
      <Latency>70</Latency>
      <PathReliability>99.0</PathReliability>
      <NumberOfDevices>2</NumberOfDevices>
      <Joins>2</Joins>
    </NetworkDevice>
  </WirelessNetworkReport>
</Response>
```

This request must be directed to the Network Manager – only the Network Manager has all of this information.

8.7.11 Returning Cached Response Messages

The following sections describe how cached response messages are returned. The following sections describe how cached response messages are returned. The HART commands which can be cached by the Gateway are summarized in Table 10.

Returning Cached Burst Mode Response Messages

If Burst Mode has been set up then the cached HART Response messages will be returned by the Gateway. If the Response message has not been cached then the Request will be forwarded directly to the device. The following HART commands that will be supported are summarized in Table 10.

The following diagram illustrates the sequence for returning cached Response messages:

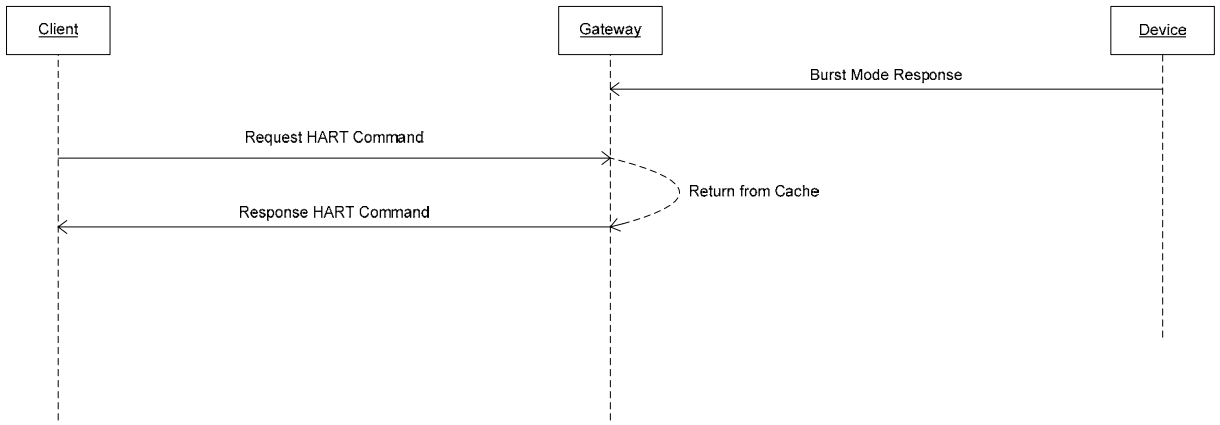


Figure 23. Cached Burst Mode Response Messages

The service request and response are summarized below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="01245">
  <HARTCommand DeviceId="12" HARTAddress="93 03 07 15 E2" Command="33" Length="4">
     00 01 02 03
  </HARTCommand>
</Request>


<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="01245" status="success">
  <HARTCommand DeviceId="12" HARTAddress="93 03 07 15 E2" Command="33"  Length="26" Status="00 10">
     00 27 41 3F E0 00 01 39 00 00 00 00 02 06 41 00 E0 00 03 39 42 49 C0 00
  </HARTCommand>
</Response>
```

Returning Cached Event Notification Response Messages

If Event Notification has been set up then the cached HART Response messages will be returned by the Gateway. If Event Notification has not been set up then an error response is returned. In this case the Client should issue Command 48.

Command 119 is slightly different from the other burst mode commands. Command 119 will continue to be transmitted from the Device to the Gateway until the alarm bits have been acknowledged. Upon receiving a command 119 the Gateway should acknowledge using command 120.

Returned Cached Read/Write Response Messages

In several cases, responses to HART commands will be cached by the Gateway. In these cases the cached Response message will be returned to the client. The following HART commands can be cached by the Gateway are summarized in Table 10.

The following diagram illustrates this sequence for returning cached Response messages:

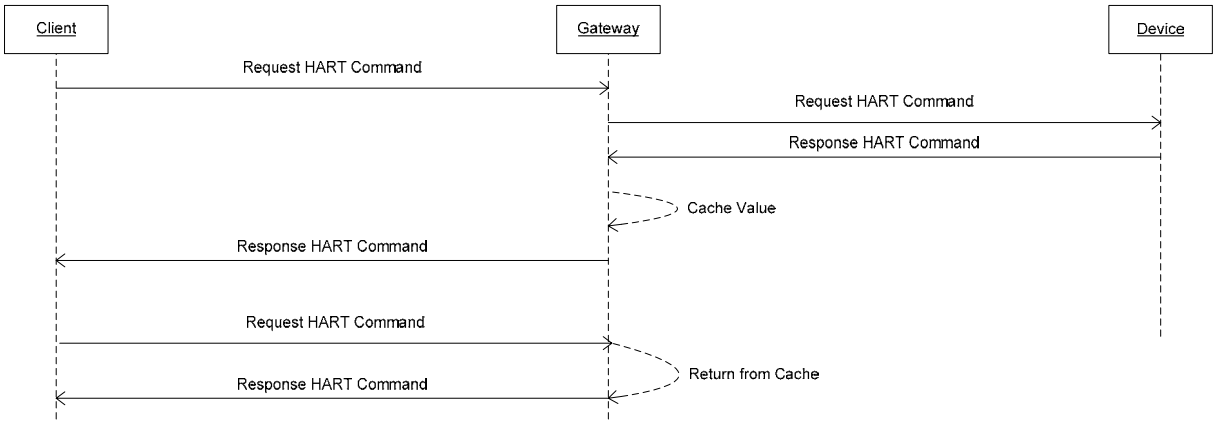


Figure 24. Returning Cached Read/Write Responses

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="01245">
  <HARTCommand DeviceId="12" HARTAddress="93 03 07 15 E2" Command="0" Length="0">

  </HARTCommand>
</Request>


<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="01245" status="success">
  <HARTCommand DeviceId="12" HARTAddress="93 03 07 15 E2" Command="0"  Length="26" Status="00 10">
     41 3F E0 00 27 41 3F E0 00 39 42 47 C0 00 06 41 00 E0 00 39 42 49 80 00
  </HARTCommand>
</Response>
```

If the client does not want the cached result it can instruct the Gateway to read directly from the device – this is indicated through the "ReadFromDevice" tag. An example is shown in the Request message below.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="01245">
  <HARTCommand DeviceId="12" HARTAddress="93 03 07 15 E2" Command="0" Length="0"
ReadFromDevice="True">

  </HARTCommand>
</Request>
```

Reading Information about the Cache Itself

The Gateway supports requests to read information about its cache. The Client can request the Gateway to return a summary for all of the devices (default). The Client can also request the Gateway to return the cached responses for one, multiple, or devices. This summary request and response are summarized below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="cactus">
  <GatewayCache summary="True"> </GatewayCache>
</Request>

<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="bluebonnet" status="success">
  <GatewayCache summary="True">
    <SourceDevice deviceId="Device2" HARTAddress="93 03 07 15 E2" >
      <HARTCommand Command="0"/>
      <HARTCommand Command="48"/>
      <HARTCommand Command="9"/>
    </SourceDevice>
    <SourceDevice deviceId="Device3" HARTAddress="93 03 07 15 E3">
      <HARTCommand Command="0"/>
      <HARTCommand Command="48"/>
    </SourceDevice>
    <SourceDevice deviceId="Device4" HARTAddress="93 03 07 15 E4" >
      <HARTCommand Command="0"/>
      <HARTCommand Command="48"/>
    </SourceDevice>
  </GatewayCache>
</Response>
```

This complete request and response for device 3 is summarized below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="cactus">
  <GatewayCache summary="False" DeviceId="Device3">
  </GatewayCache>
</Request>


<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="cactus" status="success">
```

```
 <GatewayCache summary="False">
  <SourceDevice DeviceId="Device3" HARTAddress="93 03 07 15 E3">
   <HARTCommand Command="0" Length="26" Status="00 10">
   41 3F E0 00 27 41 3F E0 00 39 42 47 C0 00 06 41 00 E0 00 39 42 49 80 00
   </HARTCommand>
   <HARTCommand Command="48" Length="10" Status="00 10">
   00 80 08 00 00 00 13 67
    </HARTCommand>
  </SourceDevice>
 </GatewayCache>
</Response>
```

8.7.12 HART Request/Response Commands sent to a Device through a Gateway

The following diagram illustrates the process in which a HART command is sent by a client to a device and the HART response is returned back to the client. In this case the Gateway routes the request to the device and the Gateway in turn routes the response from the device back to the client. The following diagram illustrates this Request / Response sequence:

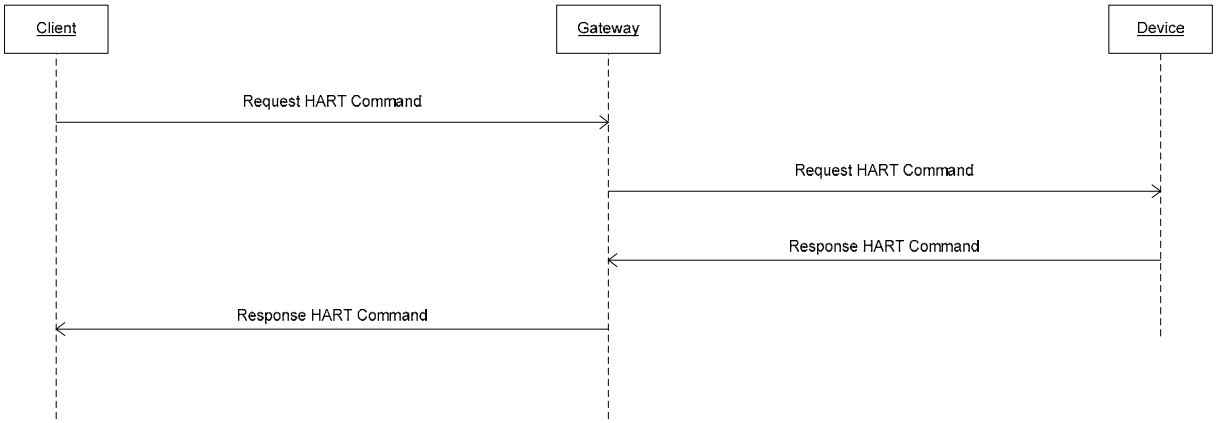


Figure 25. HART Commands sent to a Device

The service request and response are summarized below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="01245">
 <HARTCommand DeviceId="12" HARTAddress="93 03 07 15 E2" Command="129" Length="1">
    15
 </HARTCommand>
</Request>

<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="01245" status="success">
 <HARTCommand DeviceId="12" HARTAddress="93 03 07 15 E2" Command="129"  Length="7" Status="00 10">
    15 41 A0 00 00
 </HARTCommand>
</Response>
```

8.7.13 Setting up Gateway Change Notification Services

The Gateway can send unsolicited change notification messages to the 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. This service was described earlier.

The following sequence illustrates how a Client could use notification services:

1. Client1 requests change notification messages for three devices, Device1, Device2, and Device3.
2. The Gateway grants the request and sets up a change notification service for Client1.
3. Device2 sends a command 3 using burst mode 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 and clears the bit 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 XML-based service request and response message are summarized below:

Step 1 - Client1 requests change notification messages for three devices, Device1, Device2, and Device3 and for Network Topology changes.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="bluebonnet">
  <ChangeNotification leaseId="567" leasePeriod="600000">
    <DeviceMonitorRequest deviceId="Device1">
      <AttributeToMonitor>
        BurstMode
        DeviceStatus
        DeviceConfiguration
      </AttributeToMonitor>
    </DeviceMonitorRequest>
    <DeviceMonitorRequest deviceId="Device2">
      <AttributeToMonitor>
        BurstMode
        DeviceStatus
      </AttributeToMonitor>
    </DeviceMonitorRequest>
    <DeviceMonitorRequest deviceId="Device3">
      <AttributeToMonitor>
        DeviceConfiguration
      </AttributeToMonitor>
    </DeviceMonitorRequest>
    <NetworkMonitorRequest>
      <AttributeToMonitor>
        NetworkTopology
      </AttributeToMonitor>
    </NetworkMonitorRequest>
  </ChangeNotification>
</Request>
```

Step 2 - The Gateway grants the request and sets up a change notification service for Client1.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="bluebonnet" status="success">
  <ChangeNotification leaseId="567" leasePeriod="600000">
    <DeviceMonitorRequest deviceId="Device1">
      <AttributeToMonitor>
        BurstMode
        DeviceStatus
        DeviceConfiguration
      </AttributeToMonitor>
    </DeviceMonitorRequest>
    <DeviceMonitorRequest deviceId="Device2">
      <AttributeToMonitor>
        BurstMode
        DeviceStatus
      </AttributeToMonitor>
    </DeviceMonitorRequest>
    <DeviceMonitorRequest deviceId="Device3">
      <AttributeToMonitor>
        DeviceConfiguration
      </AttributeToMonitor>
    </DeviceMonitorRequest>
    <NetworkMonitorRequest>
      <AttributeToMonitor>
        NetworkTopology
      </AttributeToMonitor>
    </NetworkMonitorRequest>
  </ChangeNotification>
</Response>
```

Step 4 - The Gateway processes the change notification list for Client1. It adds command 3 for Device 2 to its change list and clears the bit for Client1.

```
<?xml version="1.0" encoding="UTF-8"?>
<Notification transactionId="bluebonnet">
    <DeviceMonitorRequest deviceId="Device2">
      <Commands>
        3
      </Commands>
    </DeviceMonitorRequest>
</Notification>
```

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

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="01245">
  <HARTCommand DeviceId="Device2" HARTAddress="93 03 07 15 E2" Command="3"  Length="0">
  </HARTCommand>
</Response

<Response transactionId="01245" status="success">
  <HARTCommand DeviceId="Device2" HARTAddress="93 03 07 15 E2" Command="3"  Length="26" Status="00
10">
     41 3F E0 00 27 41 3F E0 00 39 42 47 C0 00 06 41 00 E0 00 39 42 49 C0 00
  </HARTCommand>
</Response>
```

8.7.14 Reading Gateway Settings

The Gateway supports requests to read its internal settings. This is summarized below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="cactus">
  <GatewaySettings/>
</Request>

<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="01245" status="success">
  <GatewaySettings>
    <MaxDevices>250</MaxDevices>
    <NumberOfDevices>56</NumberOfDevices>
    <MaxBurstMode>8</MaxBurstMode>
    <NumberOfBurstMode>2</NumberOfBurstMode>
    <MaxBlockModeTransfers>2</MaxBlockModeTransfers>
    <NumberOfBlockModeTransfers>0</NumberOfBlockModeTransfers>
    <MaxDelyedResponses>2</MaxDelyedResponses>
    <NumberOfDelayedResponses>0</NumberOfDelayedResponses>
    <MaxNumberOfRetries>3</MaxNumberOfRetries>
  </GatewaySettings>
</Response>
```

8.7.15 Reading Gateway Retry Limit

The Gateway supports requests from Client applications to adjust its settings. The service request and response interaction is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Request transactionId="cactus">
  <GatewayCache summary="True">
  </GatewayCache>
</Request>

<?xml version="1.0" encoding="UTF-8"?>
<Response transactionId="cactus" status="success">
  <Configuration MaxNumberOfRetries ="3">
  </Configuration>
```

</Response>

8.8 Redundancy

8.8.1 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 must be considered:

- Routing
- Synchronization
- Fault Detection
- Switch Over

8.8.2 Routing

The Network Manager is responsible for establishing networking routing. The figure below shows the relationship of the Network Manager to the WirelessHART Network.

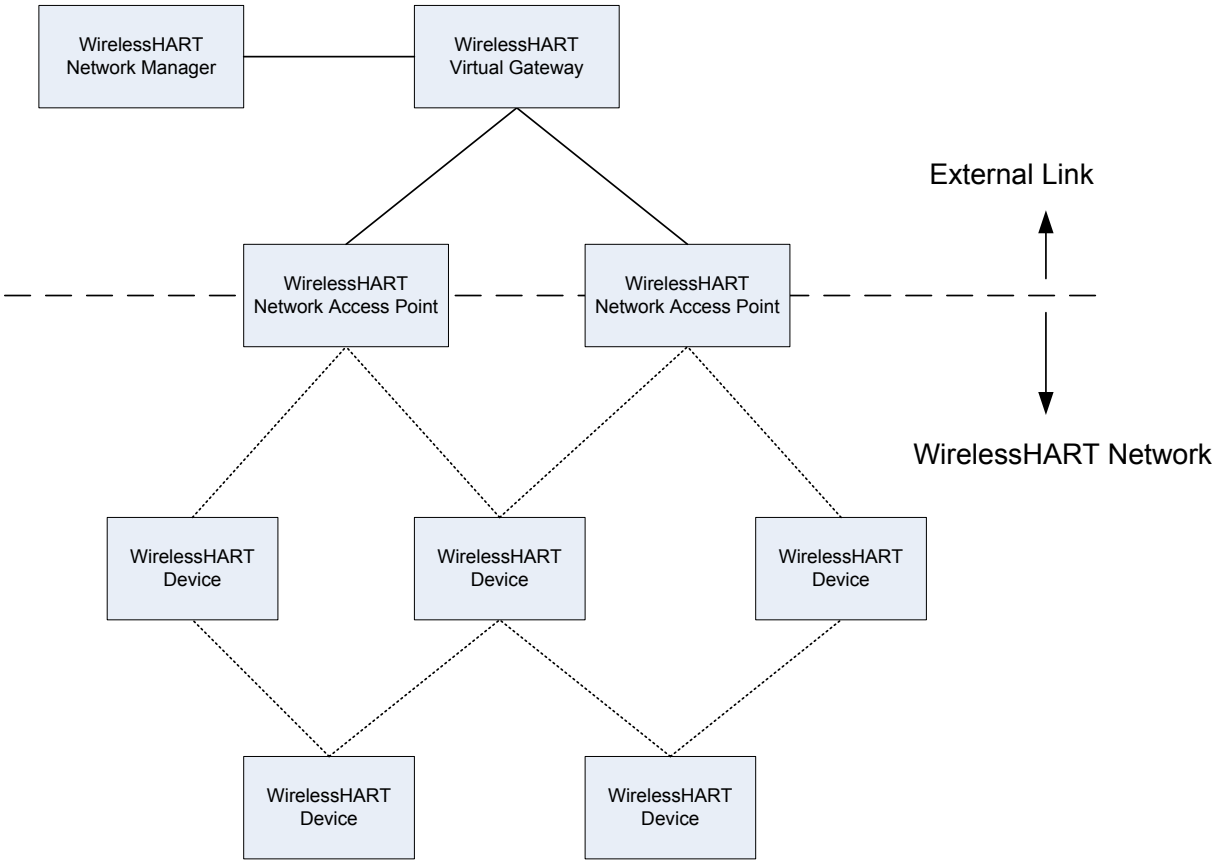


Figure 26. Network Routing

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. WirelessHART does not define this connection or the method of securing it.

All the 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 below.

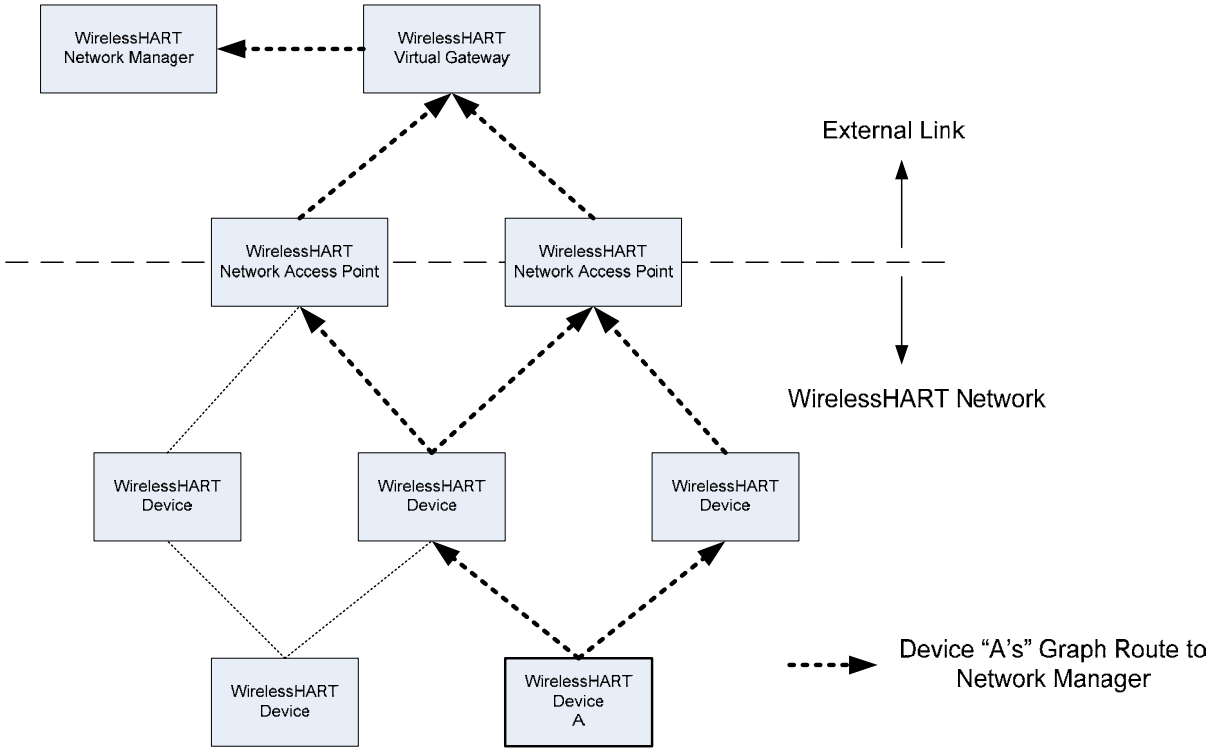


Figure 27. Graph Routing from WirelessHART Device "A" to the Network Manager

8.8.3 Network Manager Redundancy

Network Manager redundancy itself can be implemented in a variety of ways. One way to implement Network Manager redundancy is shown below. Network Manager Redundancy is outside the scope of the WirelessHART speciation.

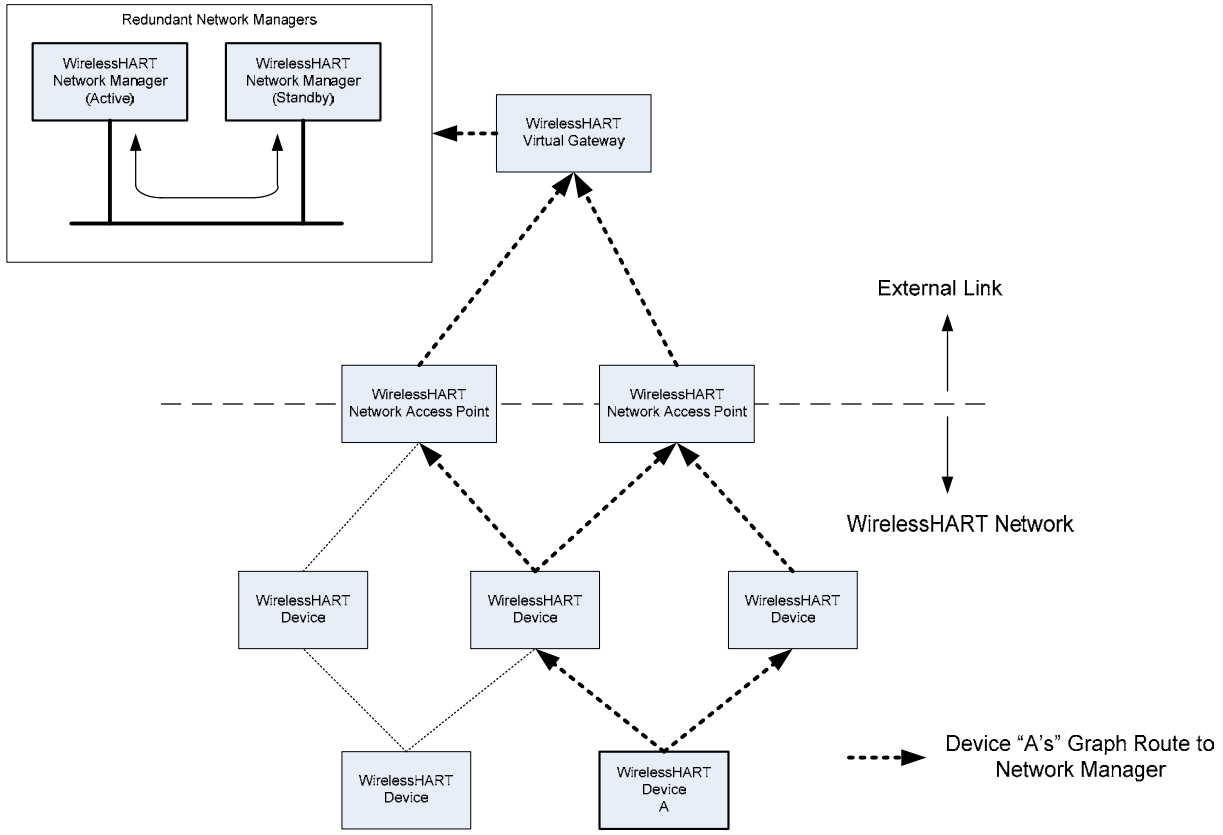


Figure 28. Redundant Network Managers

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 should 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.

9. WIRELESSHART NETWORK MANAGER

This section describes 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 joining and leaving 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 the *Network Management Specification*. The commands that the Network Manager uses to setup, monitor, and manage the overall network are described in the *Common Practice Command Specification* and the *Wireless Commands Specification*. 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 below in Figure 29.

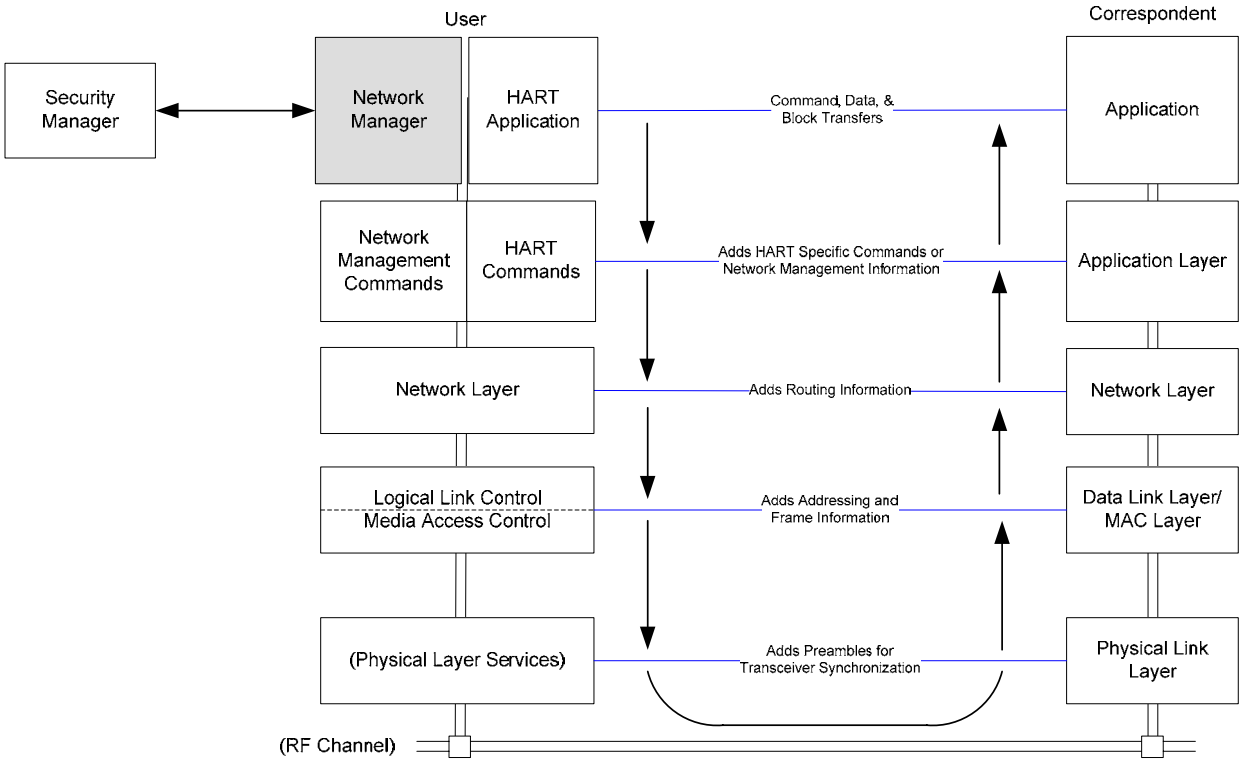


Figure 29. Network Manager Scope

So the Network Manager can 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 below in Figure 30.

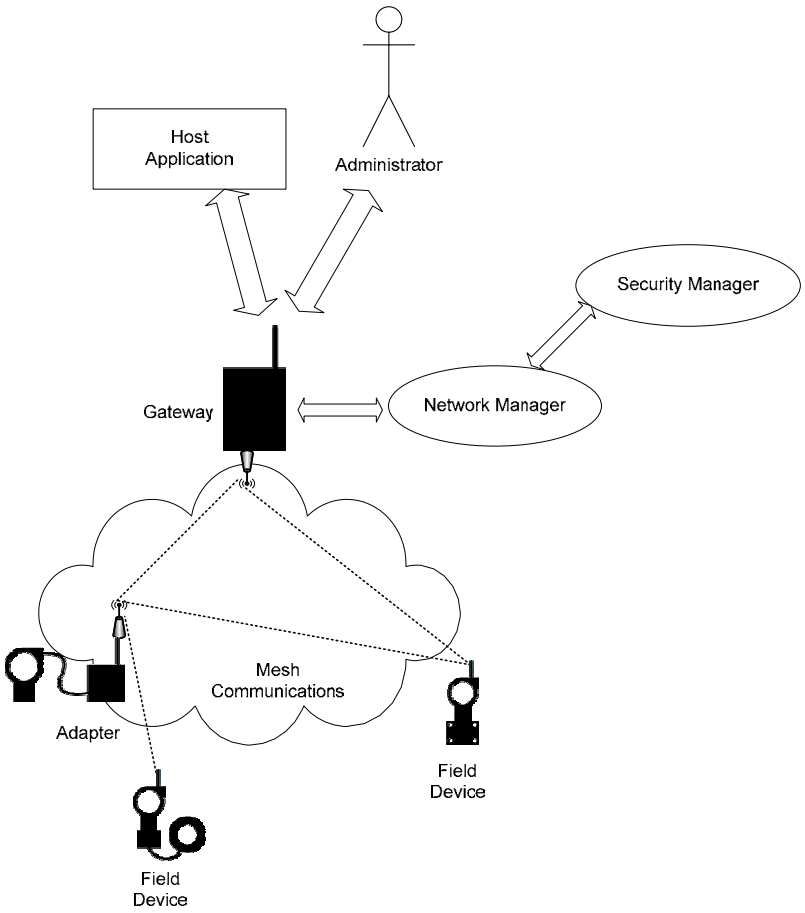


Figure 30. Network Manager in WirelessHART Network

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.

9.1 Core Network Functions

9.1.1 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 3, 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.

9.1.2 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.

9.1.3 Security Manager

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 outside the scope of this specification.

9.1.4 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 HART commands.

9.1.5 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.

9.1.6 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 must 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.

9.1.7 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 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.

9.1.8 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 must 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 must 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.

9.1.9 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 should 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.

9.1.10 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.

9.2 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 in below in Table 13.

Table 13. 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.
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.

Network Function	Requirement	
	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 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 burst mode, 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 below in Figure 38. The following sections describe the components that go into the overall Network Management Architecture. The final section brings the complete architecture together.

9.3 Network Manager Model

9.3.1 General Model for Network Manager

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 31.

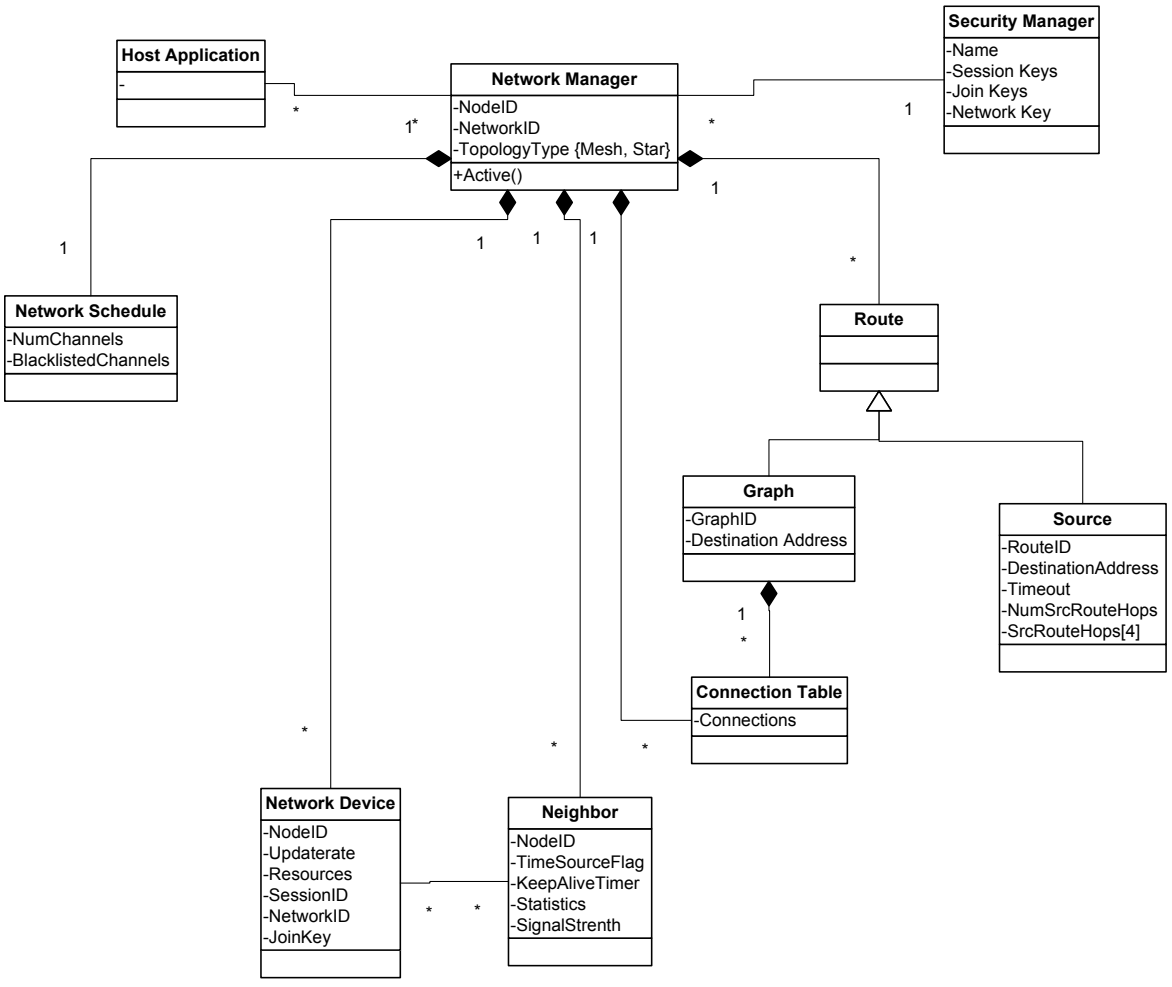


Figure 31. General Model for Network Manager

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. 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 4 neighbors 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.

Initializing the Network Manager

The Network Manager is responsible for the Network Id – the Network Id must 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 the *Network Management Specification*.

9.3.2 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 32. These network device types were described earlier in the document.

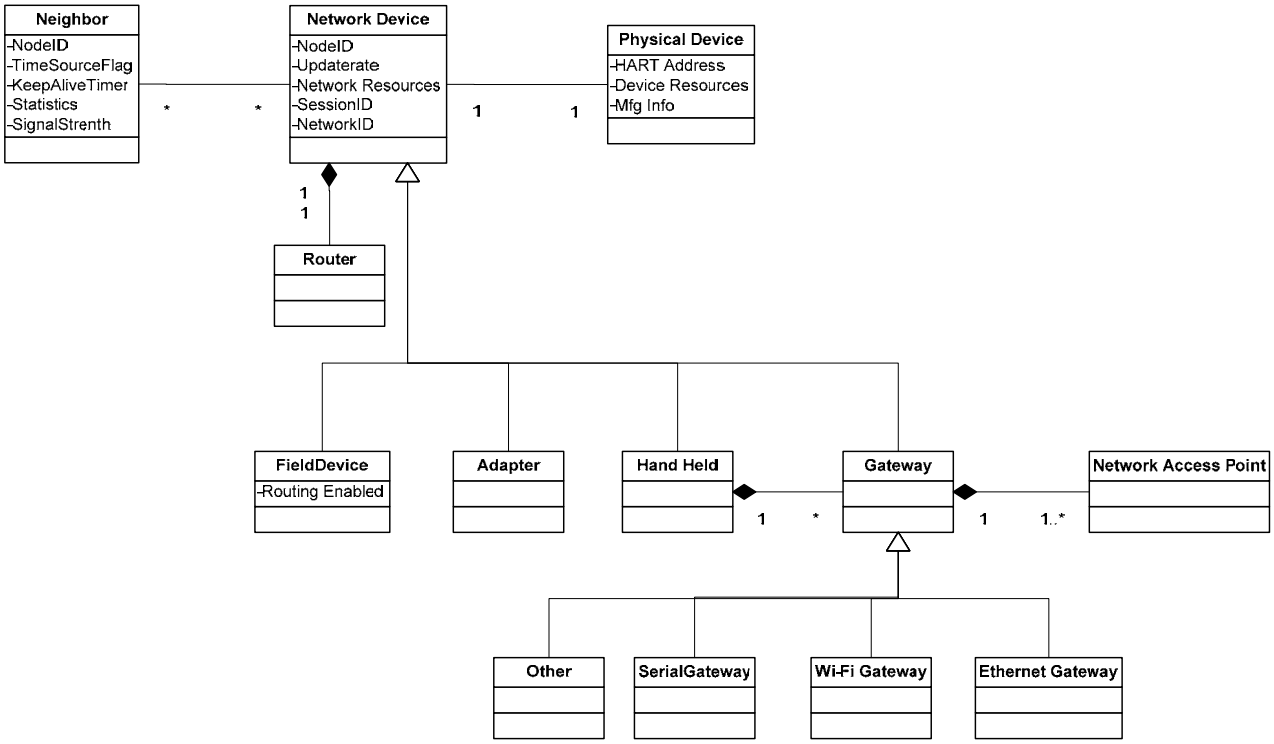


Figure 32. Kinds of Devices

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 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 must 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 FSK modem.

Routers are used to transfer messages from one location to another. Any device can act as a router.

Gateways are used to connect the WirelessHART Device network to Host applications. The Gateway connects to the WirelessHART Network through Network Access Points.

9.3.3 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 33.

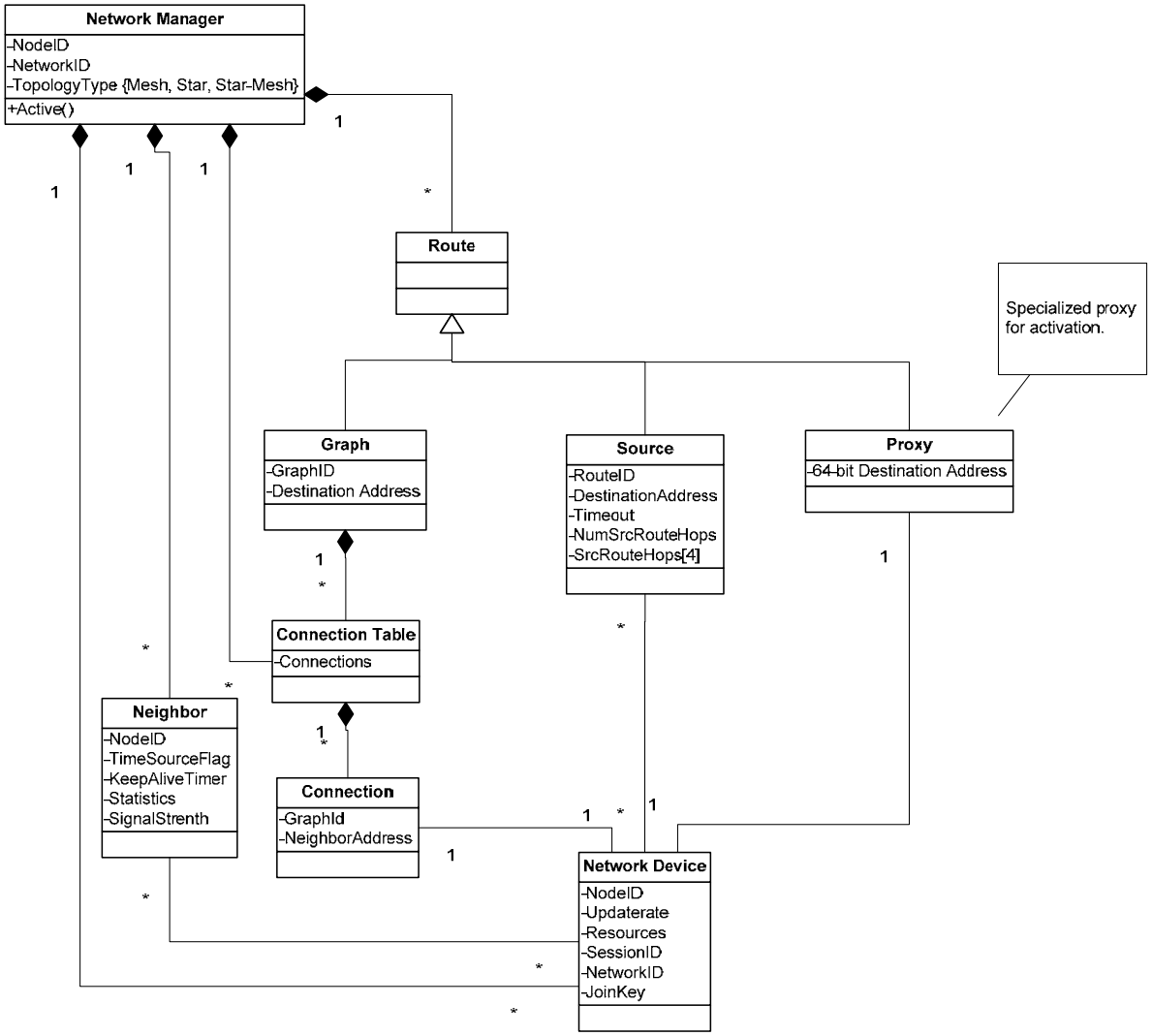


Figure 33. Network Routing

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 optiization 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

From this it 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 must 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.

9.3.4 Network Schedule

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 34.

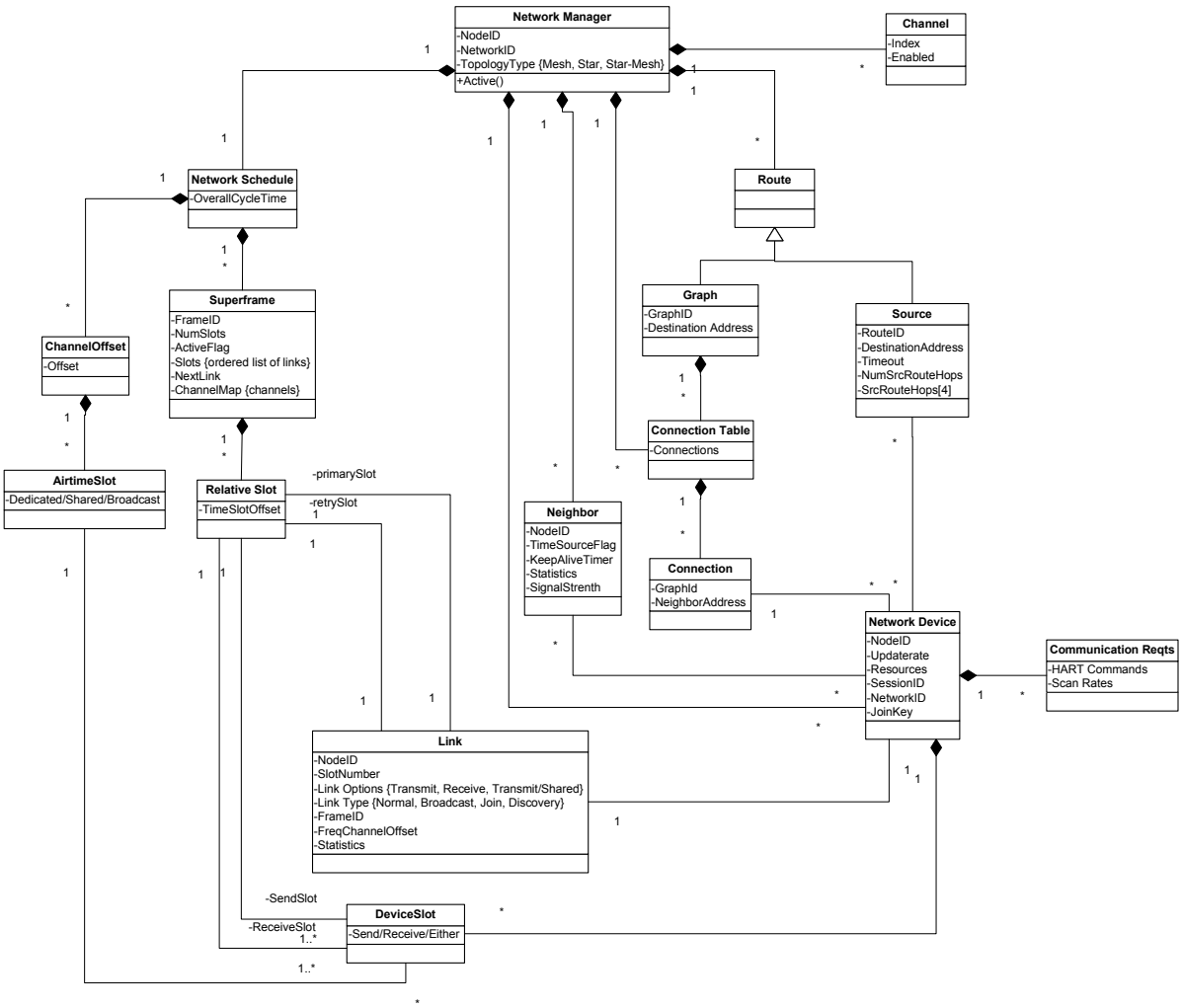


Figure 34. Network Schedule

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 should 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 opportunity to communicate. The device determines what it will communicate in each slot.

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 35 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.

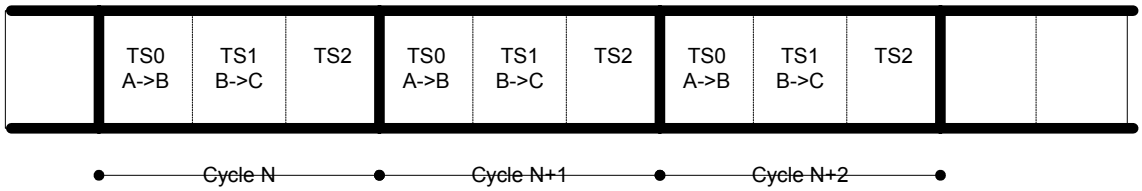


Figure 35. Example of a Three-slot Superframe

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.

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, Burst 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 applications 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 – we need to be able to do this to support asset management, diagnostic, and specialized applications that require high throughput for durations of time measured in minutes). 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 36. Because all Superframes begin at the same time, it is always possible to identify time of a given Superframe cycle and time slot.

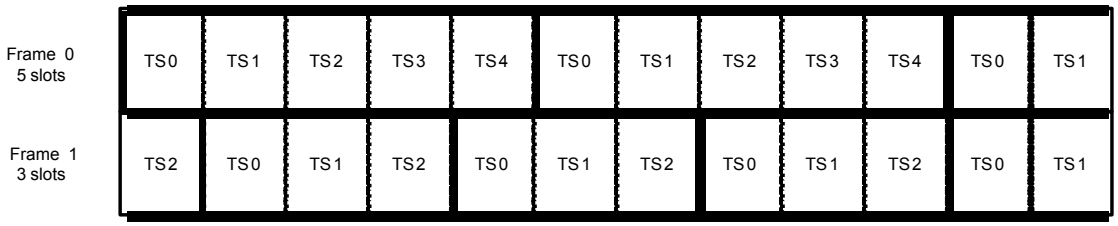


Figure 36. Multiple Superframes in a Network

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 must operate on the link that has the numerically lowest FrameId. The rules for link arbitration are defined in the *TDMA Data-Link Layer Specification*.

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 slot repeat in time at the rate corresponding to 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 must be awake and listening.

Links
When the Network Manager creates connections between devices, link assignments must 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.

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.

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.

9.3.5 Security Manager

The Security Manager is used by the Network Manger to allocate Session Keys. This is show in Figure 37.

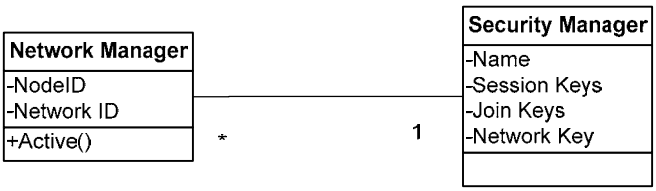


Figure 37. Security Manager

The Network Manager is responsible for propagating security keys (see the *TDMA Data-Link Layer Specification* and *Network Management Specification*).

9.3.6 Detailed Model for Network Manager

This section brings together all the pieces of the Network Manager. The Network Manager is shown in Figure 38 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 must 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.

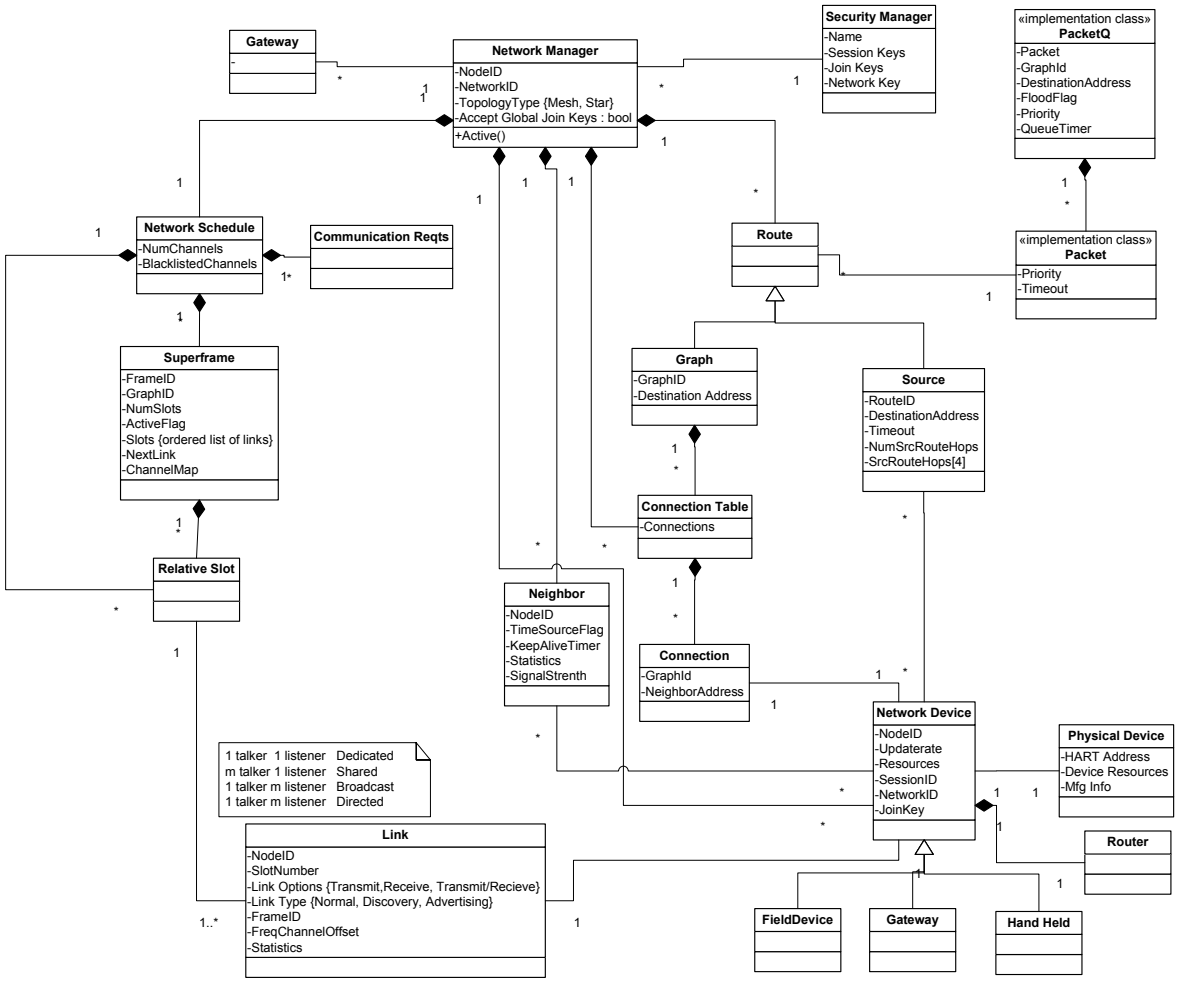


Figure 38. Network Management Architecture

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.

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 must 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 the *Network Management Specification* (HCF_SPEC-85).

9.4 Routing

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. This section summarizes some of the key requirements, presents rules for determining routes, and outputs a simple example of a route.

9.4.1 Routing Requirements

The requirements for the Network Manager are summarized in the table below.

Table 14. 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.
Manage 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 must be generated.
Downstream graphs from the Virtual Gateway to all of the nodes in the network must be generated.

9.4.2 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" 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.

9.5 Scheduling

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. This section summarizes some of the key requirements, presents rules for determining routes, and outputs a simple example of a route.

9.5.1 Schedule Requirements

The requirements for developing the schedule are summarized in Table 15.

Table 15. 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 must 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 250msec, ½ 500msec, 1 sec, 2 sec, 4 sec, 8sec, 16 sec, 32 sec, and 60 sec. (or more)
	Base Network Management and Burst Mode communications should not exceed 30% of the available communication bandwidth (100 slots/sec max).
	Services – the network manager must 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.

Function	Requirement	
	From the furthest end device, allocate one link for each en-route Network Device to the Gateway. Allocate a 2nd dedicated slot on the same path to handle a retry. Allocate a 3rd 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 6400 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 must 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).
		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 must 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 second (two links in each direction).

9.5.2 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 burst mode data first.
- Burst mode 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 burst mode data 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 (6400 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 3 slots every 15 minutes for health reports.
- Each device needs to have at least 1 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 second in duration – there are 4 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 400ms).
- Alternate the slots as XMIT, RECEIVE (should all be Shared).

Optimization

- Number of hops to the gateway
- Alternate paths
- Latency
- Power utilization
- Overall throughput

9.5.3 Networking Scheduling Example

In this section a simple scheduling example will be presented. The example covers a WirelessHART Network that consists of 1 Gateway and 3 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 seconds. The arrangement is shown in Figure 39.

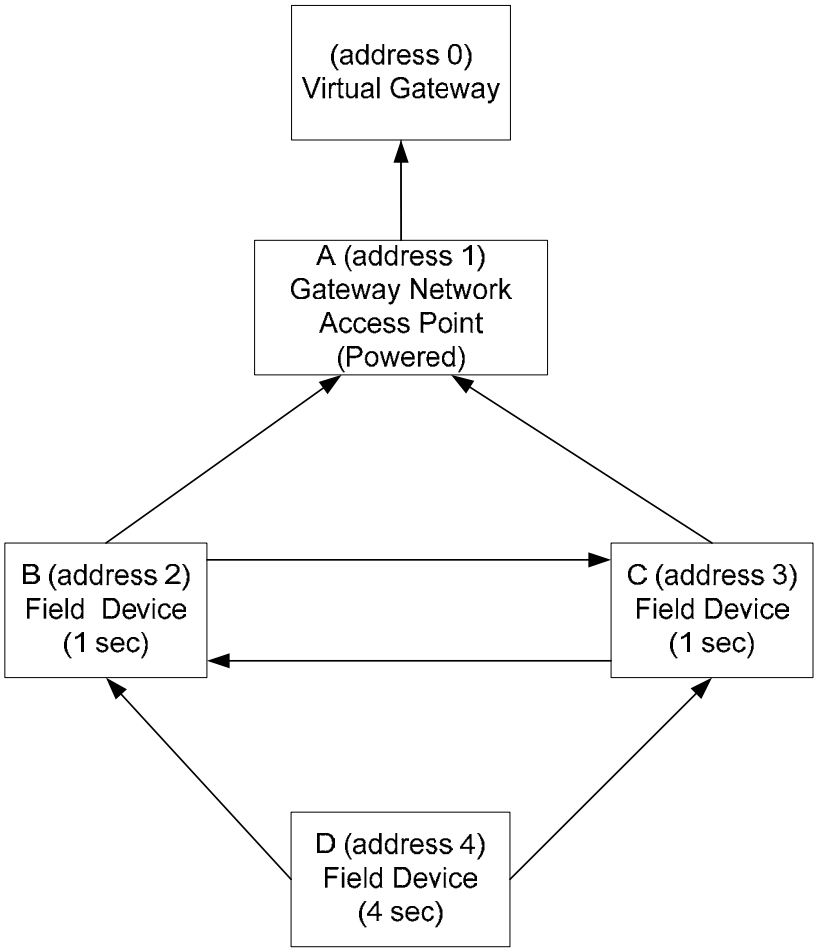


Figure 39. Example Four Network Device WirelessHART Network

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

FrameId 1: 1 Second 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									

FrameId 4: 4 Second 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

FrameId 0: Management Superframe (Superframe Length 6000 – once per minute)

Advertisements (advise joining devices about open links to talk on)

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8
0									*->A
1							*->B		
2	*->C								
3	*->D								
4									

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									

Join Response (shared w/ management requests)

Ch Offset	…	TS11	TS12	TS13	TS14	TS15	…		
0			A->B	A->C	A->B				
1						B->D			
2									

Commands

Ch Offset	…	TS16	TS17	TS18	TS19	TS20	…		
0									
1			B->A	B->A					
2		C->A							
3		D->B							
4									

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)

Device 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

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

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

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

9.5.4 Process Control Example

An example for a bioreactor is shown in Annex C. The example includes several measurements, several regulating valves, and several blocking valves. The elements included in the schedule are summarized below.

Table 16. Bioreactor Example

Category	Measurement
Measurement	Reactor Level (LT210)
	Feed Flow (liquid –FT201)
	Reactor Gas Pressure (PT208)
	Reactor Temperature (TT207)
	Agitator Amps (IT209)
	Return Water Temperature (TT206)
	Reagent Flow (FT203)
	Air Flow (FT202)
	Dissolved Oxygen (AT205)
	pH (AT204)
Regulating Valve	Feed Flow (FV201)
	Reagent flow (FV203)
	Coolant Flow (FV206)

Category	Measurement
	Vent Flow (FV208)
	Air flow (FV202)
Blocking Valve	Charge Flow (FZ211)
	Harvest Flow (FZ212)
	Harvest Flow (FZ213)

9.6 Network Manager Interface

The Network Manager communicates to Network Devices through a series of Wireless Commands (see *Wireless Command Specification*). 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 must implement all of the Data Link Layer and the Network Layer Commands listed in the *Wireless Command Specification*.

Each HART 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 40 shows two message sequences. In the first, 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.

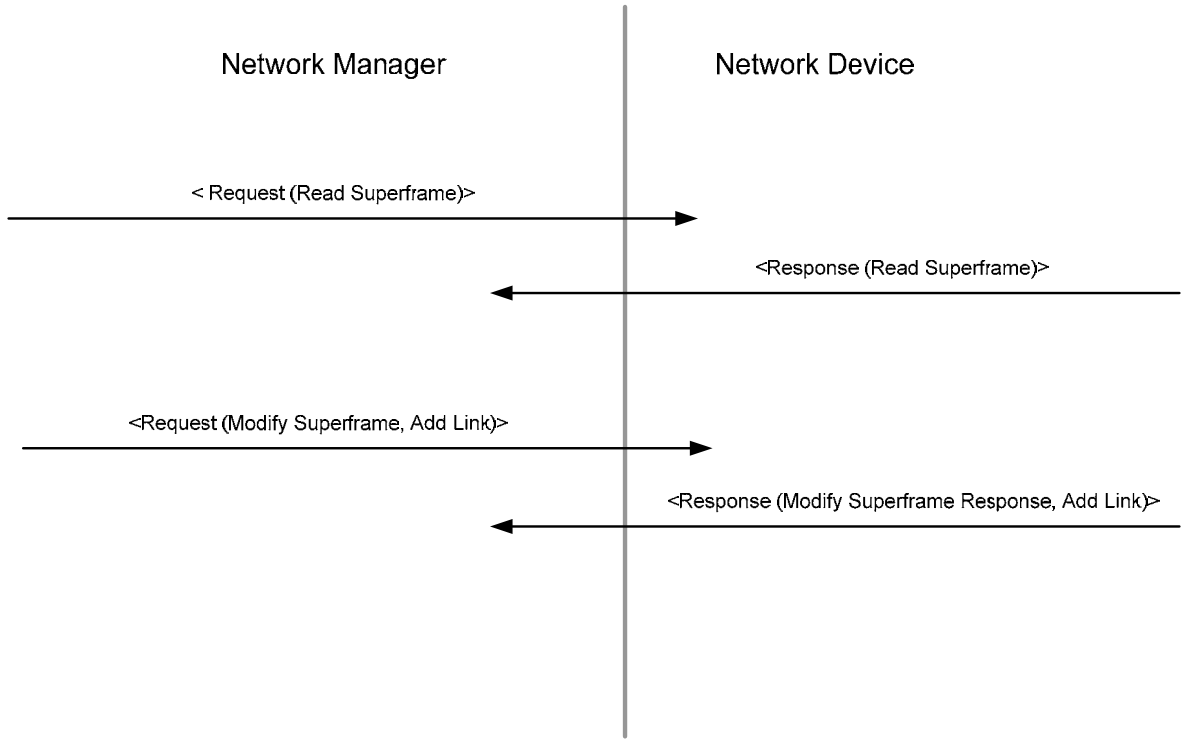


Figure 40. Examples of HART Command Message Sequences

Scenarios describing how to use the Network Manager interface are summarized below.

Table 17. 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.

9.6.1 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 must exist and they must 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 below in Figure 41.

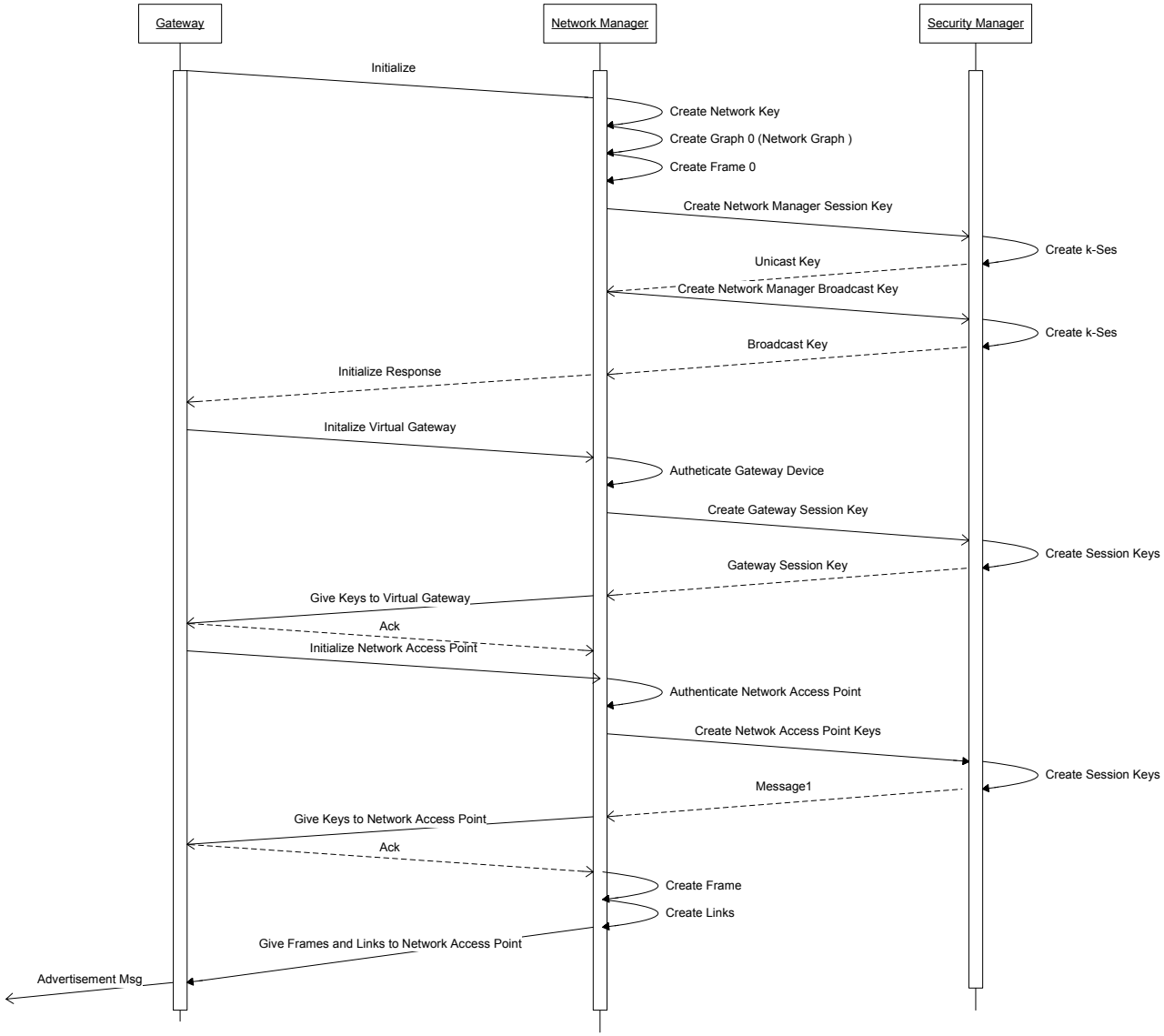


Figure 41. Initializing a WirelessHART Network

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.

9.6.2 Allocating Communication Resources

A device that joins the network should 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. 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 Specification*).

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 42.

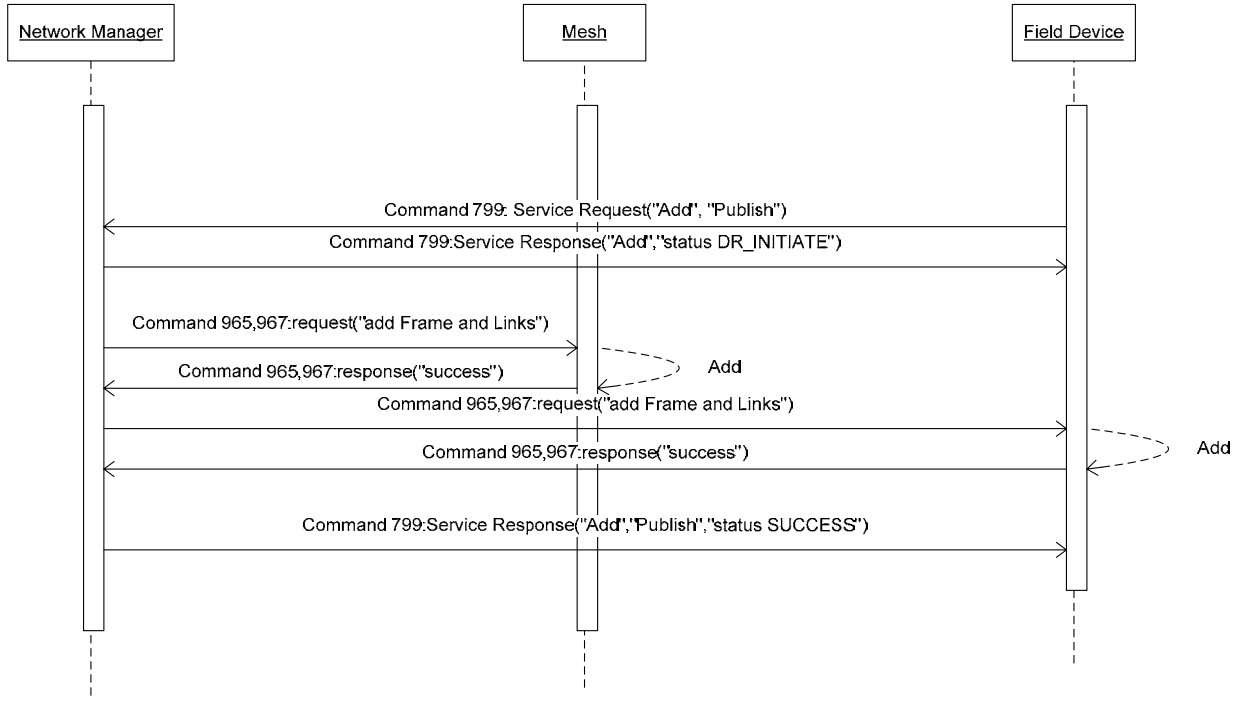


Figure 42. Allocating and using services

9.6.3 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 43.

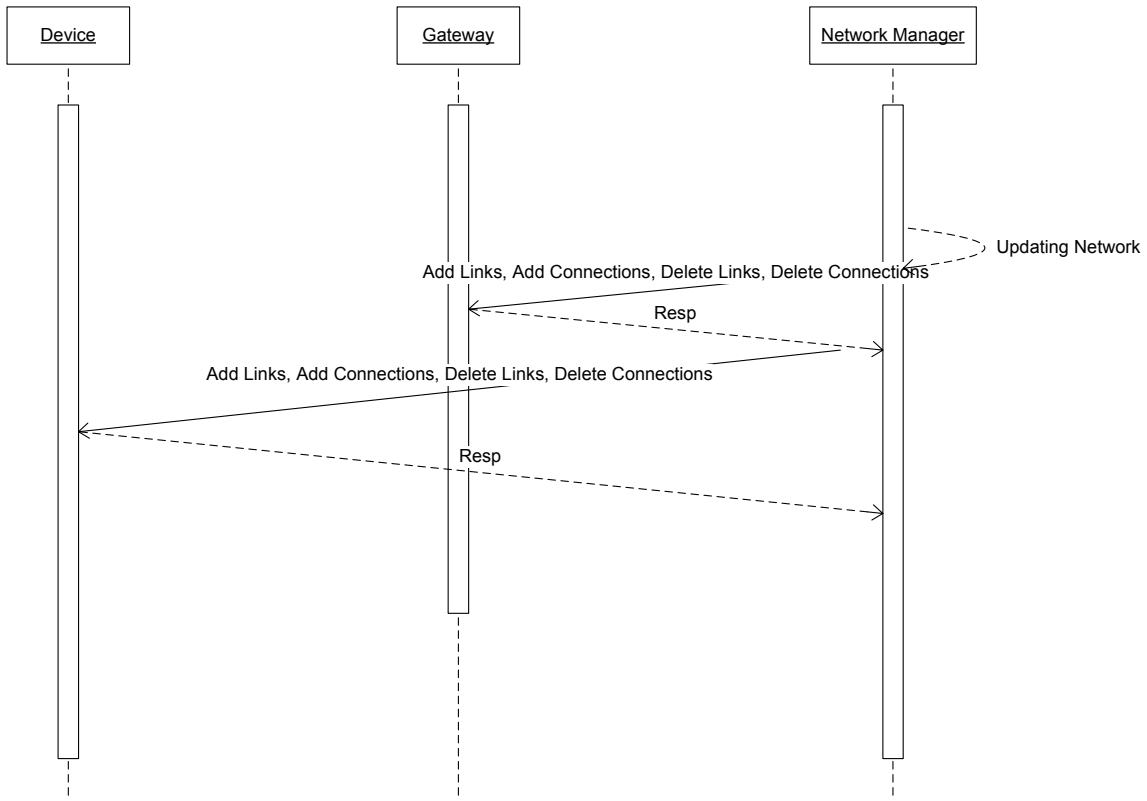


Figure 43. Adjusting Network Schedule

9.6.4 Health Reports

Health reports are transferred from Network Devices to the Network Manager periodically. The Helath reports include These messages are not acknowledged. This is illustrated below in Figure 44. Schedule and network optimization should be performed periodically as well.

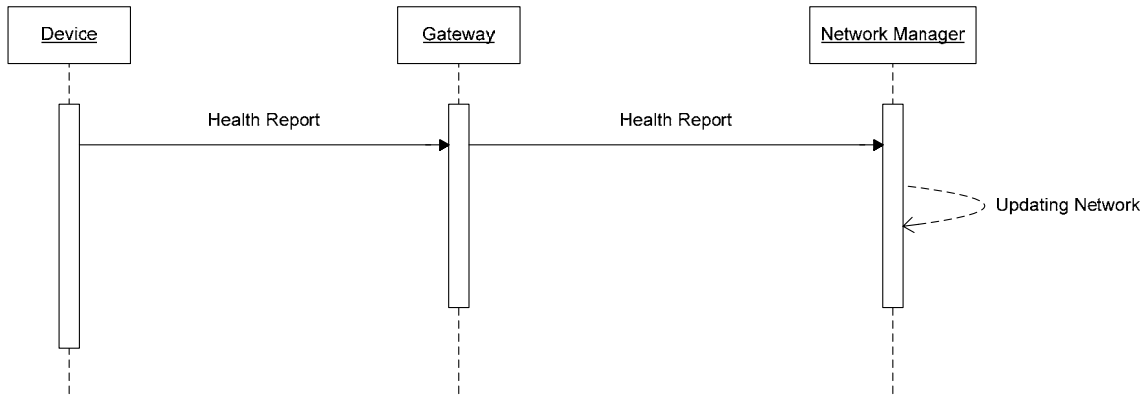


Figure 44. Health Reports

10. HANDHELDS

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. **HART 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. **HART Handheld connected through an FSK modem on the device**—In this mode the Handheld Device connects through an FSK modem directly to the 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 45.

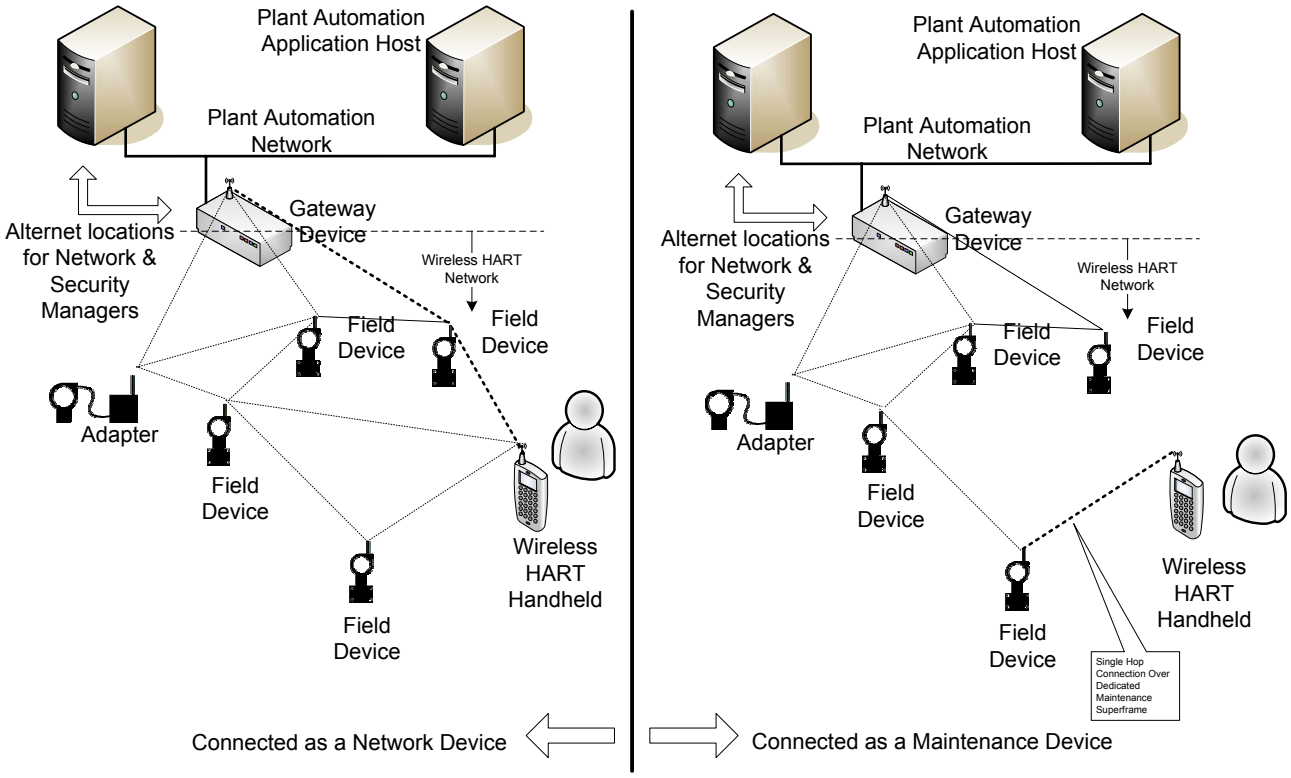


Figure 45. WirelessHART Handheld Connections

10.1 General Requirements

The WirelessHART Handheld must comply with all requirements for Host Conformance Class 3 - "Generic Host" (see the *Command Summary Specification*). In addition, the Handheld should meet all the requirements of Conformance Class 5 - "Universal Host" through the use of DDL. The handheld must meet all WirelessHART requirements.

10.2 Wired HART Interface

On the Token-Passing Data-Link the WirelessHART Handheld is a master. Functioning as a master device allows the handheld to configure devices.

All attributes and commands supported by the Field Device are available via the wired interface and must be supported by the handheld. When requested, the Field Device answers Identity Commands normally thus allowing the handheld load the Field Device's DD.

When connected via the wired interface the handheld does not have access to the wireless network.

10.3 WirelessHART Handheld Connected as a Network Device

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 (note - initializing the network and the gateway are implementation specific). 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.

10.3.1 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.

10.3.2 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).

10.4 WirelessHART Handheld Connected as a Maintenance Device

Once a network up and running maintenance technicians will want to be able to connect up to individual devices in the field to gather information, run diagnostics, calibrate, etc. In order to support these activities the WirelessHART Network 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 must to talk to (See Subsection 10.3).

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.

ANNEX A. WIRELESSHART GATEWAY IMPLEMENTATIONS

A.1. Scope

This section describes possible WirelessHART Gateway implementations.

A WirelessHART Gateway contains the computation resources, communication interfaces, and physical entities to support the functions and features described in this document. It is envisioned that a variety of Gateways may be developed that fit into several different tiers, each with distinct differences in capability and functionality. For instance, the lowest tier of Gateways may be simple devices that provide a single RS-232/485 port that allows connectivity to a PC-based application. On the other end of the scale, a large multi-purpose Gateway may have several communication ports, such as Serial Modbus, Profibus and Ethernet TCP/IP, and have embedded applications such as web browser, real-time database, cached data, graphics, etc.

An actual WirelessHART Gateway may be implemented as a single physical box with a port providing client access, for example a TCP port. Or it may also be implemented using several physical boxes – each box serving as a separate access points into the WirelessHART Network. Although a specific WirelessHART Gateway implementation is not defined by the WirelessHART standard, single box and multiple box implementations are included below as examples.

A.2. WirelessHART Gateway with integrated Network Access Point

In a single network Access Point implementation all of the functionality of the WirelessHART Gateway, the Network Access Point, the Network Manager, and the Security Manager can be packaged together into the same physical package. This is shown below in Figure 46.

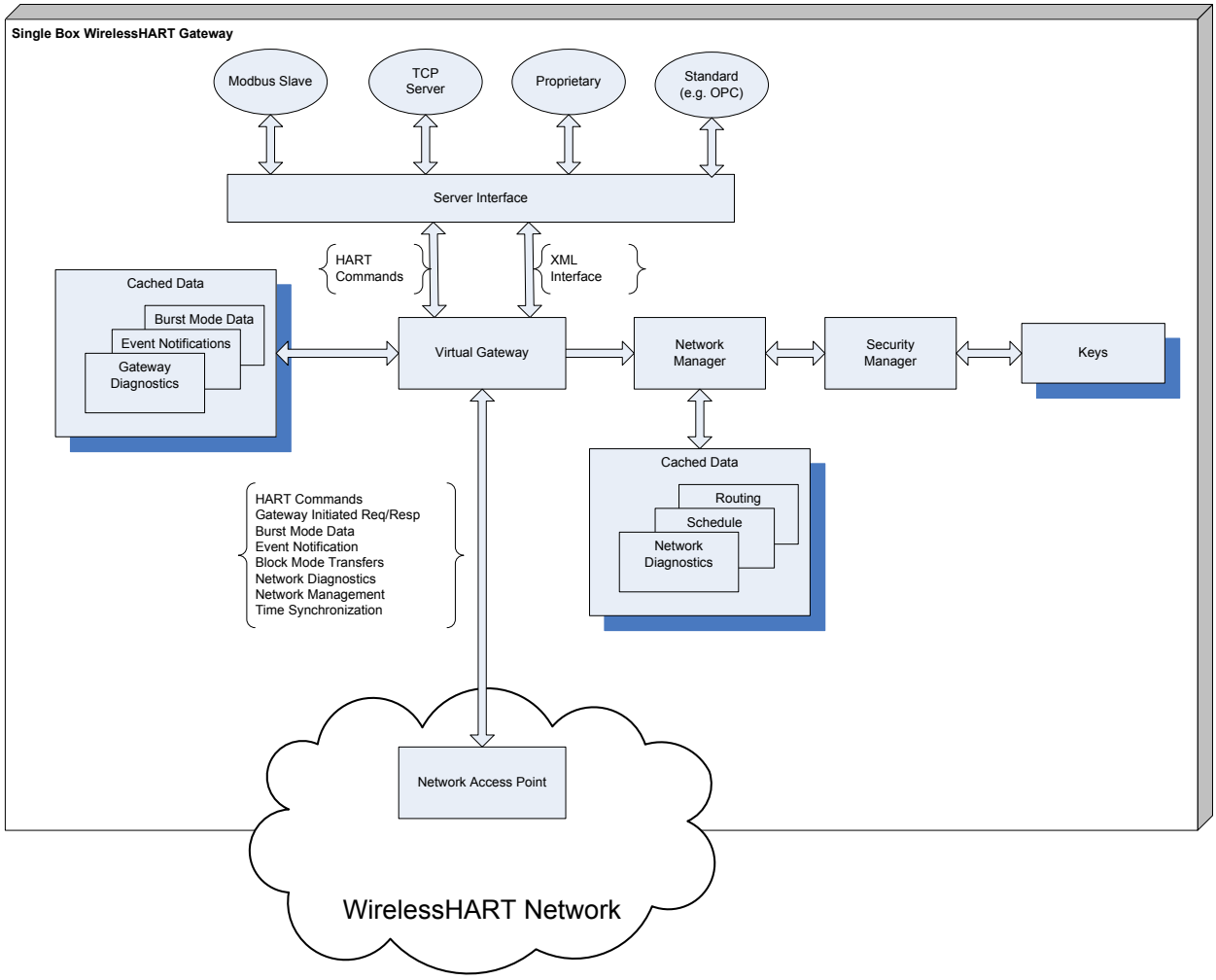


Figure 46. Single Box WirelessHART Gateway Deployment

In this arrangement network Access Point provides access to the radio and provides the WirelessHART communication stack. It may also implement most of the HART protocol – this is up to the implementers. The HART Command and XML interfaces are described in detail in later sections in this document.

A.3. WirelessHART Gateway with multiple Network Access Points

In a multi-Network Access Point implementation the features of the WirelessHART Gateway, the Network Access Points, the Network Manager, and the Security Manager can be separated. In this arrangement, multiple Network Access Points provide multiple access points into the WirelessHART Network. The Host Interface, data caching, Network Management, and Security Management can be located in a single package, multiple package, or combined with one of the Network Access Points. One possible implementation is shown below in Figure 47.

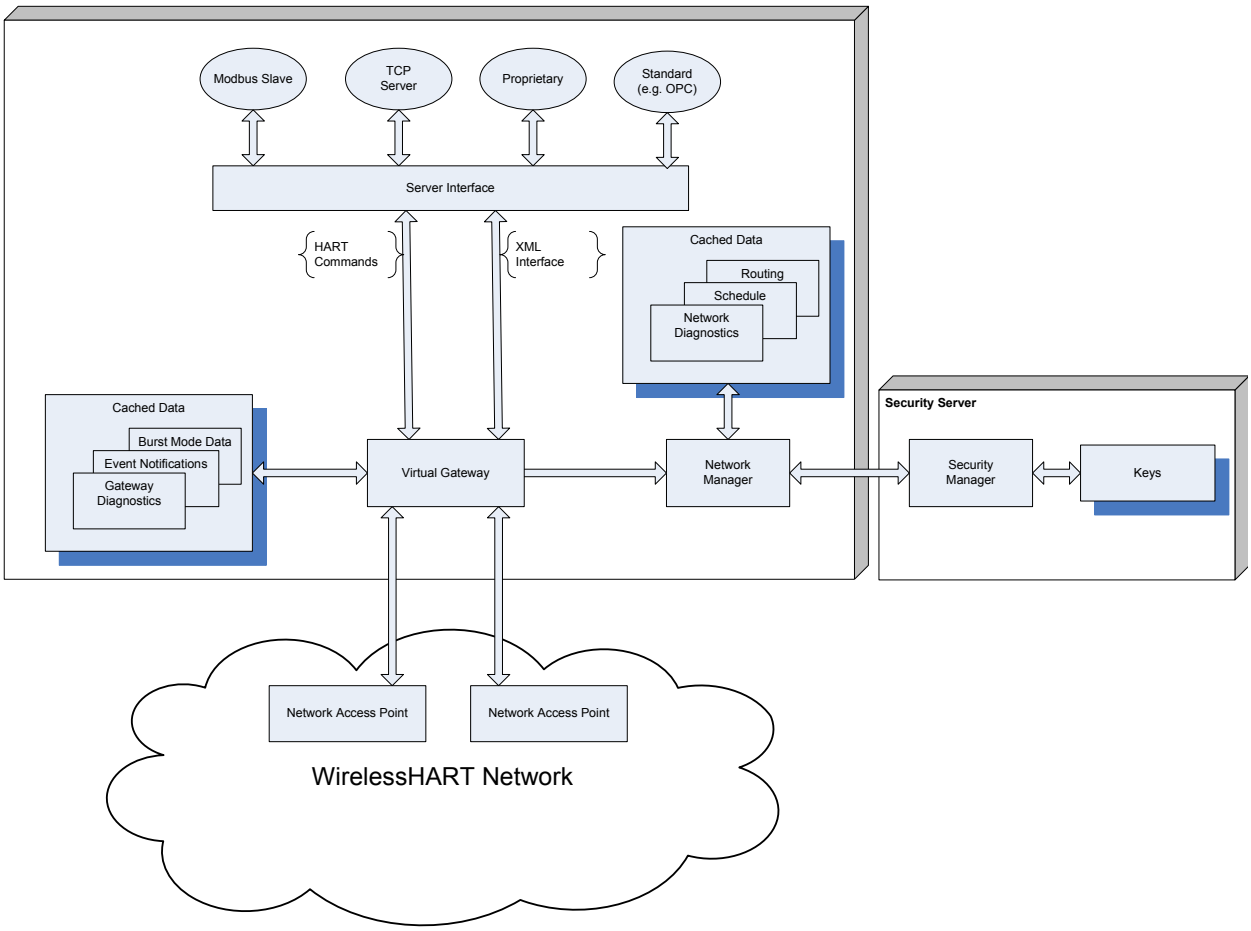


Figure 47. Multi-Box WirelessHART Gateway Deployment

In this multi-package example the Virtual Gateway, Network Access Points, and Security Manager are in separate physical packages. It is important that the connection between the Gateway and the Security Server be over a secure connection.

ANNEX B. WIRELESSHART GATEWAY XML SCHEMA

B.1. Scope

This section describes XML Schema for the Gateway.

To be completed later.

ANNEX C. SCHEDULING WIRELESSHART FOR MONITORING AND CONTROL

C.1. Scope

Many of the desirable features of the WirelessHART Network such as self-healing, self-organization, and redundant routing are achieved through the establishment and updating of a network communication schedule. The Network Manager is responsible for the creation of this schedule and the associated connections. It is also responsible for the distribution of this schedule to the individual network devices. This scheduling function may be broken into the following phases:

1. Support devices joining the network. As part of this the Network Manager is responsible for authenticating and orchestrating the join process.

2. Establishment of routes. As part of this the Network Manager is responsible for the creation of routes that can be used by plant automation hosts, gateways, other devices, and the Network Manager itself to perform communications with the application layer in network devices.

3. Schedule communications. As part of this the Network Manager is responsible for the establishment of Superframes and Slots that the user layer application of a network device may use to transfer process data, alerts, diagnostics and other traffic to the gateway for access by the plant automation host. The Superframes also includes slots for network management and the join process.

4. Scheduling control functions. For network devices that are actuators, interlocks, or any device that affects the process, the Network Manager is responsible for the establishments of Routes, Superframes, and Slots that the plant automation host may use to send setpoints and outputs to the user layer application in field devices.

5. Speeding up maintenance functions. To support high data requirements the Network Manager establishes a specific Superframe for communication between network devices and asset management and diagnostics applications. This superframe contains shared slots that may be used by any network device to transfer data from the device to the host.

6. Adapting the network. The Network Manager will continually adapt the network. The Network Manager continually collects data from devices on the health of connections and traffic patterns and uses this information to adjust routing and scheduling.

When a device is added to a network it is necessary to establish Network Management communications with the network device before it is possible to send or receive information from the user layer application within the device. This write-up describes how this may be done.

The effectiveness of the overall network ultimately boils down to a combination of routing and scheduling. The services provided in the protocol stack allow network communications to be established in many ways. In this section, one means of addressing scheduling is detailed using an example of wireless transmitters and actuators associated with a bio-reactor process as illustrated below in Figure 48.

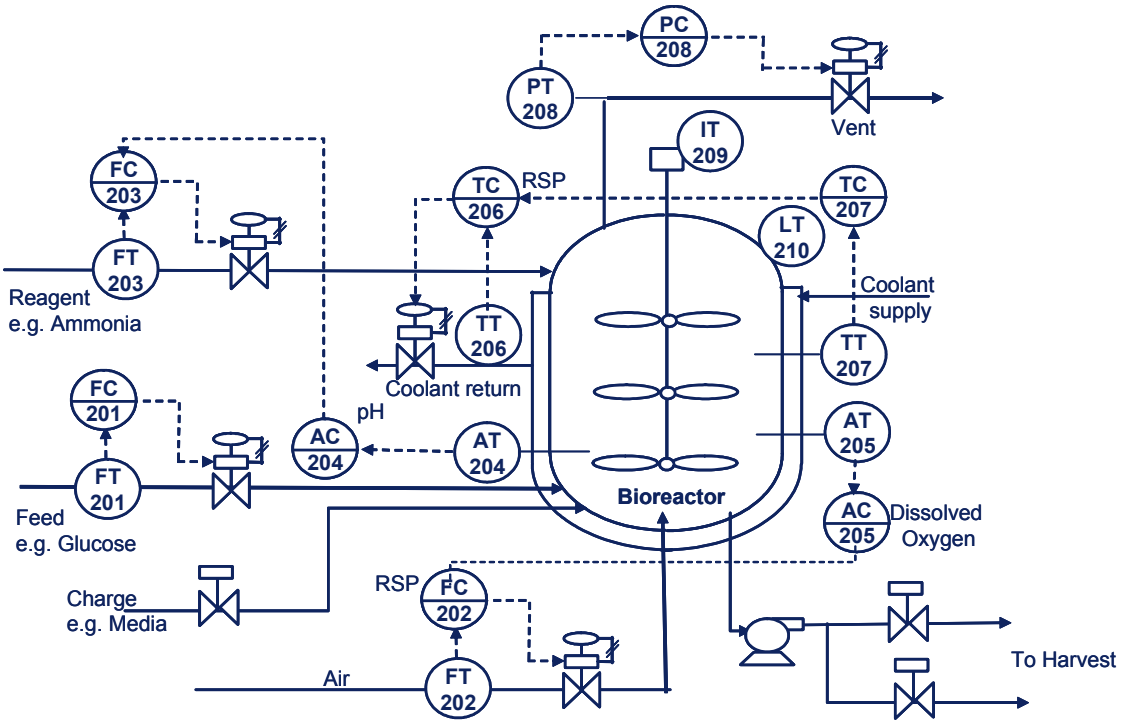


Figure 48. Bio-reactor process

The measurements, actuators, and blocking valves included in this example are summarized below:

Table 18. Instrument and Valve List for Bio-reactor

Category	Device	Measurement
Measurement	C1	Reactor Level (LT210)
	C2	Feed Flow (liquid –FT201)
	C3	Reactor Gas Pressure (PT208)
	C4	Reactor Temperature (TT207)
	C5	Agitator Amps (IT209)
	C6	Return Water Temperature (TT206)
	C7	Reagent Flow (FT203)
	C8	Air Flow (FT202)
	C9	Dissolved Oxygen (AT205)
	C10	pH (AT204)
Regulating Valve	A1	Feed Flow (FV201)
	A2	Reagent flow (FV203)
	A3	Coolant Flow (FV206)
	A4	Vent Flow (FV208)

Category	Device	Measurement
	A5	Air flow (FV202)
Blocking Valve	B1	Charge Flow (FZ211)
	B2	Harvest Flow – FZ212
	B3	Harvest Flow – FZ213

C.2. Network Management and Host Request

The network management application will normally be part of the Gateway device (it could be run in a separate host). The connection between the Network Manager and the gateway device is implementation specific. Since, a network may only have one Network Manager (which can be made redundant) and more than one Gateway Network Access Point (for improved throughput and reliability more than one Network Access Point should be used), the initial step in forming a network will be the creation of the Network Manager. There are many ways to ensure that only one Network Manager will be created – for example one of the redundant Gateways could be designated to initialize the Network Manager or a user could be required to make the selection. In any case an initialization routine will be called at the application layer that causes several things to happen:

- The initial Network Manager and all of its associated resources are created.
- The Network Manager forms a connection to the Security Manager.
- The network key is created.
- The initial Gateway is created.
- The initial Gateway is designated as the time source.
- The Network Manager creates the initial Superframes and Routing – the first order of business at this point for the Network Manager will be to form the network.

From the perspective of any field device that would like to join a network, the gateway appears as any other field device. The Protocol stack of the gateway will be the same as other network devices with modifications in the application layer for interfaces for Network Management and the plant automation host.

After the Network Manager creates the first Gateway, then the Network Manager will create the initial "Management" Superframes and Routing. Separate receive and transmit Superframes will be established and reserved for network management transmissions to network devices and for the receipt of confirmations. The Superframe ID numbers assigned to these management Superframes will be selected to be the larger than other Superframe ID's defined for the network. The management transmit Superframe will initially only include two slots that are configured for the Gateway device; a transmit slot that that is configured to periodically send Advertisement packets and a transmit slot that is reserved for the advertising device to transmit to a device that has not yet joined the network. The management receive Superframe will contain one slot for the Network Manager to receive join requests from a device that has not joined the network. After these initial network management Superframes are transferred to the gateway, then the gateway device will begin to periodically transmit Advertisement packets that contains information on the NetworkID,

time information, and slots and Superframes that have been established for a new device to communication with the gateway before it joins the network. At this point the initial Graph for the network will appear as shown below in Figure 49.

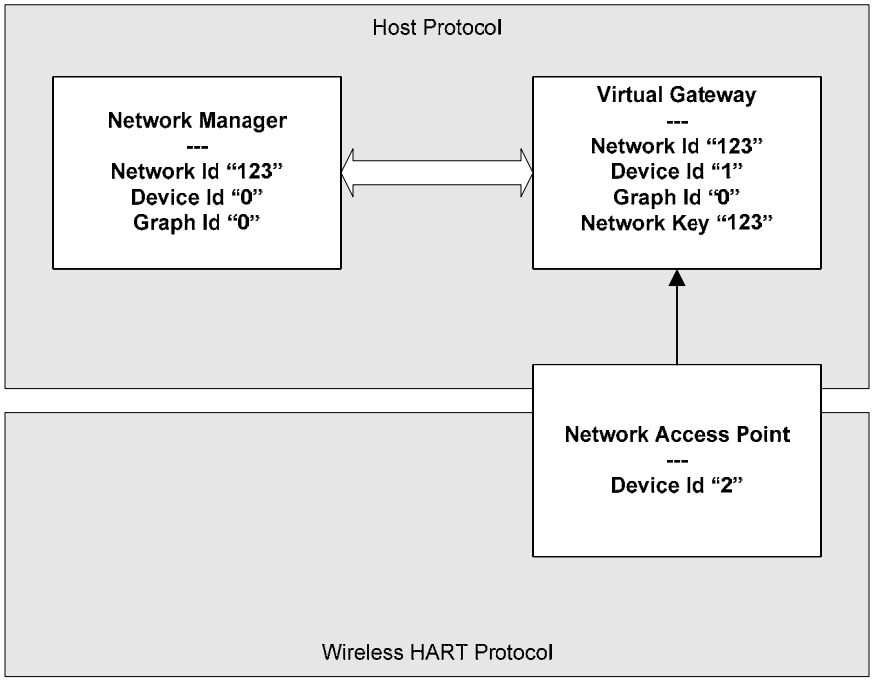


Figure 49. Initial Network Graph

The Network Manager has a well known Device ID. The Virtual Gateway is always the top of the Network Graph. In this way when additional Network Access Points are added to the network the routing to the Virtual Gateway and to the Network Manager can be maintained. This is illustrated below in Figure 50.

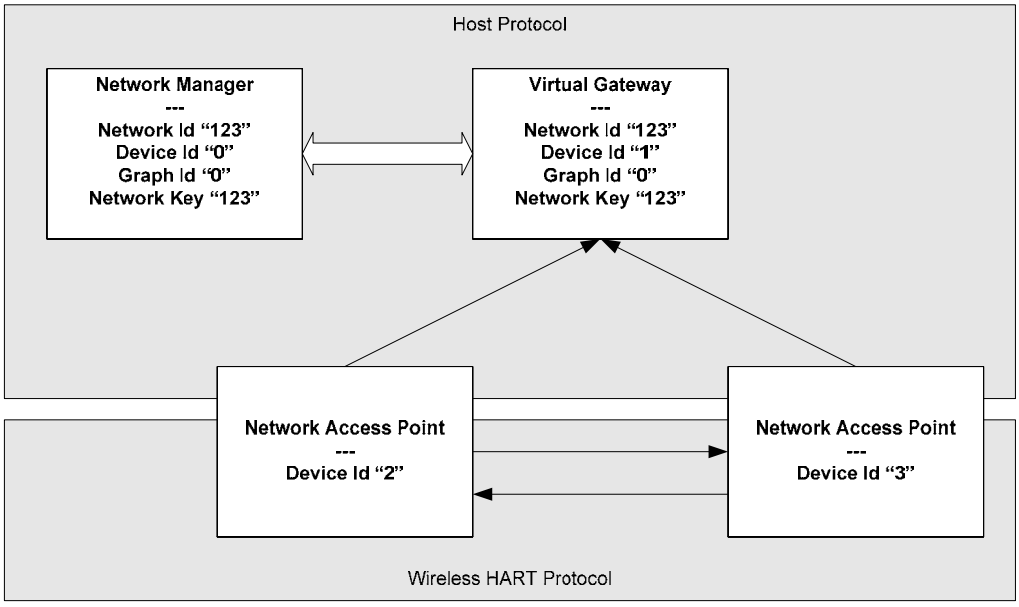


Figure 50. Adding a second Network Access Point

In this way it will always be possible to reach the Network Manager, regardless of how many Network Access Points are in use. Network Management Redundancy is described later on this document.

When a join response is received by the Gateway, the Network Manager verifies the device's *join* key5 and either completes or rejects the join process. As part of this process, the Network Manager creates dedicated slots in the management Superframe for device management and advertisement functions. A slot will be defined in the management Superframe for the newly added device to send advertisement packets. Also, shared slots will be reserved in the management Superframes for the new device to send and receive messages from a device responding to its advertisement packet. Thus, the management Superframe will contain the following information for any network device that has joined the network i.e. is active.

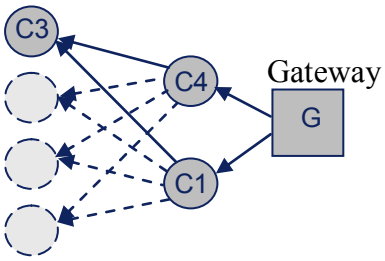
- Slot dedicated for the transmission of Advertisement packets on a periodic basis.
- Transmit and receive slots for network device(s) that joined the network through this device.
- Shared transmit and receive slots reserved for any new device that wishes to join the network through this device.

This new slot information is transferred into the network and reflected in all of the effected devices.

It is expected that any new device will listen for a period of time for advertisement packets and select the device with the strongest signal strength for joining the network. In most cases a device will be able to transmit a join request to the Network Manage since a slot is dedicated for a join request. However, if two new devices happen to transmit a join request at exactly the same time using the advertised slot and channel, then the messages will collide. In this event, collision will be detected and transmission attempted after a backoff time. Thus, the join request will be spaced and will most likely be successful on the next attempt.

Network devices join the network one at a time. Thus, overall network routing will be determined by the Network Manager based on signal strength as determined by physically layout of the plant, number of hops, and traffic flow. The overall routing will be further adjusted to reflect diagnostic information, retries, etc. In establishing a schedule for management communications, it may be assumed that only one outstanding network management communications will be allowed. Thus, a device with multiple peers may be scheduled within the same slot since the transmission will only go to one device. This allows the number of slots required for scheduling and the associated power to be reduced. Scheduling of the response should take into account the maximum response time of the devices included in the network. For example, assuming a maximum response time of 0.5 seconds, the two management Superframes may be illustrated as shown below in Figure 51 for the case where four devices have joined a network. The Network Manager is not shown in these next few drawings.

5 The only network devices that know the join key are the Network Manager and the Field Device.



Device	Peer
C3	C4, C1
C4	G
C1	G

Network Management - Transmit Frame/Graph

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8	TS9	TSn
0	G Advertise	G=>*				G=>C4	C4=>C3				
1	C4 Advertise	C4=>*					G=>C1	C1=>C3			
2	C1 Advertise	C1=>*									
3	C3 Advertise	C3=>*									

Network Management - Receive Frame/Graph

Ch Offset	TS0	TS1	TS2	TS3	TS55	TS56	TS57	TS58	TS59	TSn
0				*=>G	C3=>C1	C4=>G				
1				*=>C4			C3=>C4	C1=>G		
2				*=>C1						
3				*=>C3						

* Temporary slot reserved for communication with a new device.

Figure 51. Network Management Frames

The Network Manager is responsible for distributing the network management Superframe to the Network Devices. Only the portions of the Superframe that are directly associated with a specific Network Device will be transferred to that Network Device, as illustrated below in Figure 52 for device C4 in the previous example.

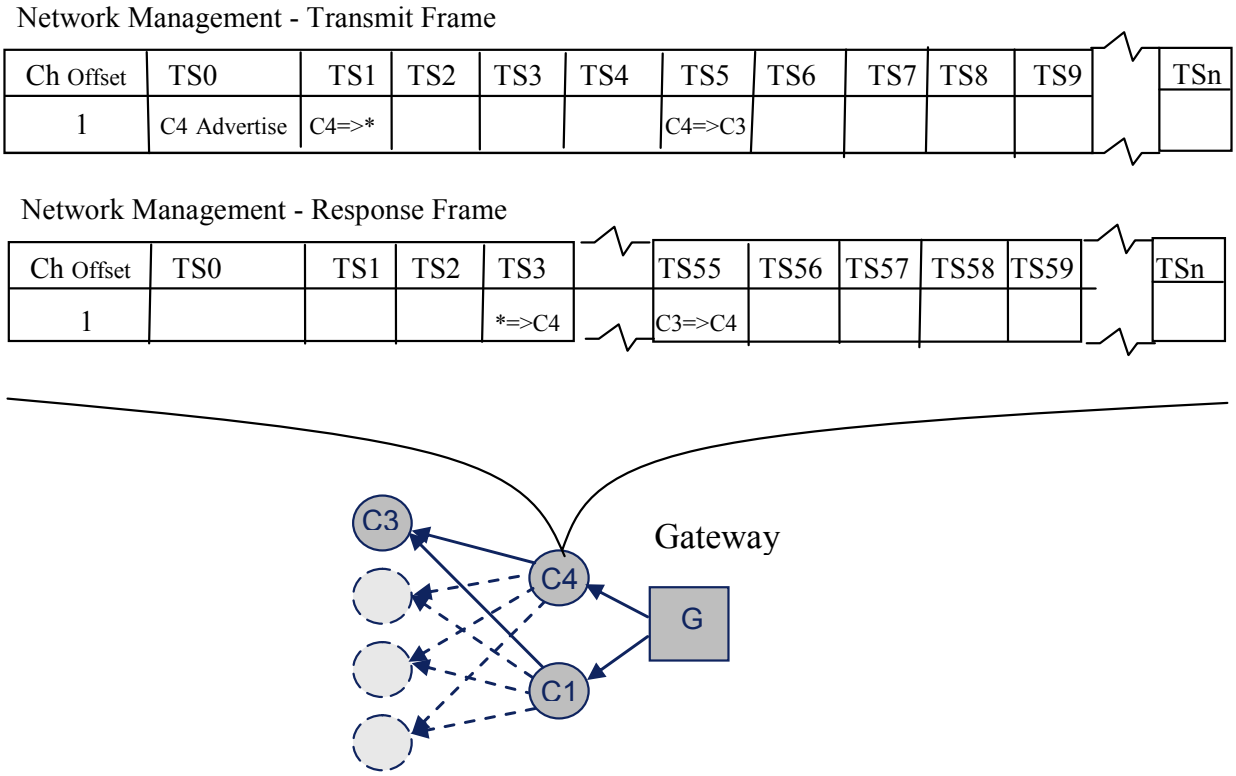


Figure 52. Network Management Frames/Graph Transferred to Device C4

The speed of response in transferring network management request and response is primary determine by the how often transmissions are scheduled to and from the Network Managers. In this example, the frequency of request can be regulated through the adjustment of the Superframe size. For example, if the Superframe size is set to 100 slots, then requests and responses will be transferred at a maximum rate of once per second with one retry (based on each slot requiring 10 ms). If the scheduled network slot communications conflicts with another Superframe, then the network management communications will automatically be delayed to avoid collisions since the network Superframe ID will always be set to be the highest numeric value.

C.3. Process Measurement

The primary objective of most field devices will be to provide a process measurement. The frequency at which this information is required by the process automation host is specific to the process equipment and the measurement type e.g. pressure, temperature, flow, level, and analytical. Thus, as part of the process automation host configuration, the user will configure the following information for all network devices that are accessed through the wireless gateway:

- Device Tag – which uniquely identifies the device e.g. HART Tag
- Measurement value(s) that are to be accessed in the network device
- How often each measurement value is to be communicated to the gateway.

For batch bio-reactor example, the field device measurements may be configured as follows:

Table 19. Measurements & Scan rates for Bio-reactor

Device	Measurement	Update Rate
C1	Reactor Level (LT210)	16 sec
C2	Feed Flow (liquid –FT201)	1 sec
C3	Reactor Gas Pressure (PT208)	1 sec
C4	Reactor Temperature (TT207)	4 sec
C5	Agitator Amps (IT209)	8 sec
C6	Return Water Temperature (TT206)	16 sec
C7	Reagent Flow (FT203)	1 sec
C8	Air Flow (FT202)	1 sec
C9	Dissolved Oxygen (AT205)	4 sec
C10	pH (AT204)	4 sec

To support the configuration of multiple Superframes for the transfer of process information at different rates, the configured measurement scan rates used by the field device and the associated communication should be configured as integer multiples of the fastest update time that will be supported by field devices. For this example, the supported scan rate will be defined as 2^x where x is positive or negative integer values e.g. scan rate selections of 1 sec, 2 sec, 4 sec, 8sec, 16 sec, and 32 sec.

To avoid introducing latency into the measurement value that is communicated to the Gateway, it is important that the processing of the sensor be coordinated with the slot configured for the measurement transmission, as illustrated in Figure 53 below. This is coordination is outside of the scope of this document.

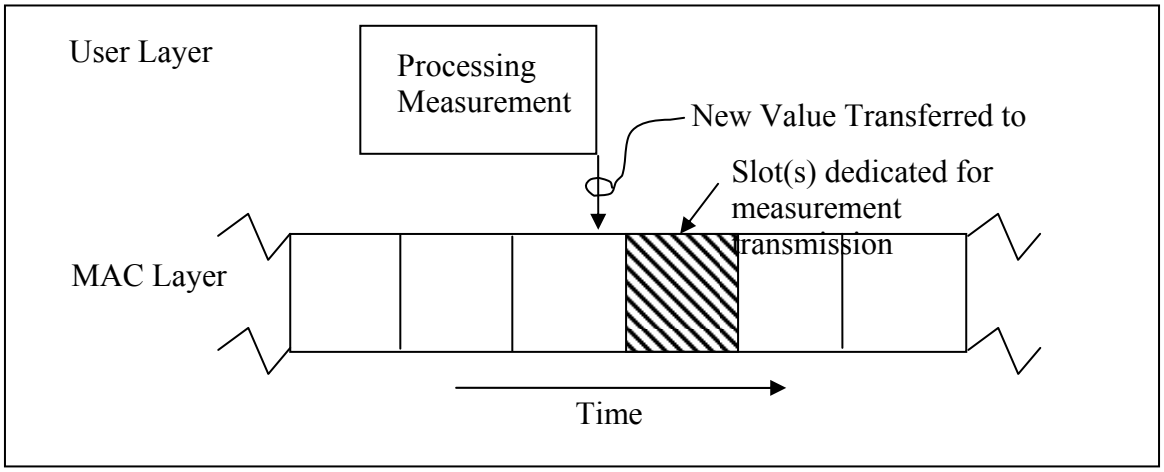


Figure 53. Synchronization of measurement processing and transmission

The scheduling of communications associated with process measurements included in a network can be simplified by defining a Superframe for each scan period and developing the schedule by allocating slots for transmission of measurement data starting with the fastest to the slowest scan rates. In the schedule, a device may only occur once within a slot time since at any given time a device is either transmitting or receiving on one channel.

C.4. Scheduling Example – Single Hop

Using the recommended approach of defining Superframes for each scan period and allocation of slots starting with the fastest to the slowest measurement, the graph associated with the batch bioreactor measurements would appear as shown below, assuming the Gateway and all the network devices are only separated by 1 hop.

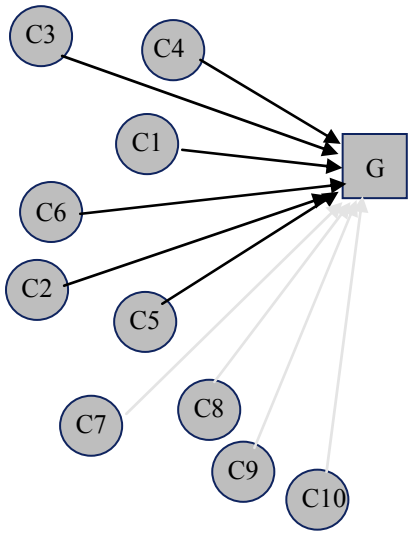


Figure 54. Batch Bioreactor Example – Single Hop

The schedule for this configuration is illustrated below in Figure 55.

Frame 0 – 1 Second Update

Ch Offset	TS0	TS1	TS2			TS99
0	C2=>G	C3=>G				

Frame 1 – 4 Second Update

Ch Offset	TS0	TS1	TS2	TS3	TS4		TS399
0			C4=>G				

Frame 2 – 8 Second Update

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6		TS799
0				C5=>G					

Frame 3 – 16 Second Update

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8		TS1599
					C1=>G	C6=>G					

Figure 55. Use of multiple frames to support different update rates

To support immediate re-transmission (if required) after a failed transmission to the gateway, additional slots would be added in the schedule immediately after each transmission. Since the Superframe size is selected to match the associated scan rate, the measurement values will be communicated as scheduled with no conflict in slot usage, as illustrated below in Figure 56.

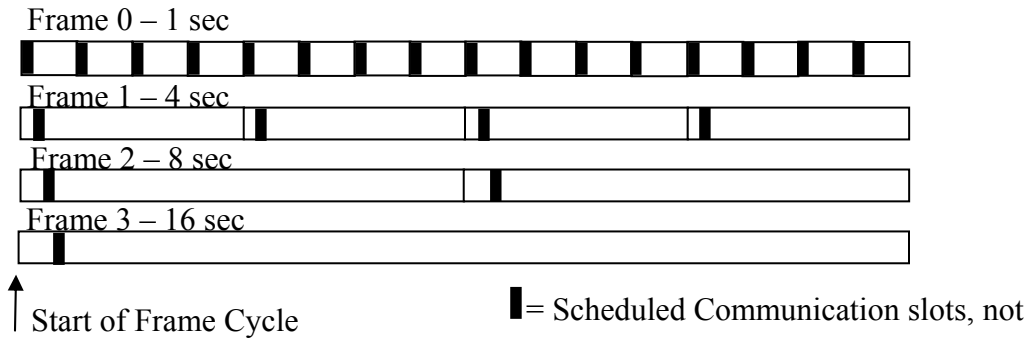


Figure 56. Adding additional slots to improve reliability

It is not expected that use of duplicate slots would be required except in the case of a super critical measurement. In most cases the retry mechanisms that are built in with Graph Routing are sufficient to ensure that a data packet is successfully transferred from the source Network Device to the Gateway.

An alternate scheduling approach to that described above would be to only define one Superframe for the transmission of process measurements where the Superframe length is set based on the shortest or longest measurement scan rate. The disadvantage of using a single Superframe with length base on either the fastest or slowest scan rate may be summarized as follow:

One Superframe - Length Based on Fastest scan rate - For measurement made at a slower rate, the associated transceivers would be powered up as often as the update rate of the fastest measurement and thus increase power consumption

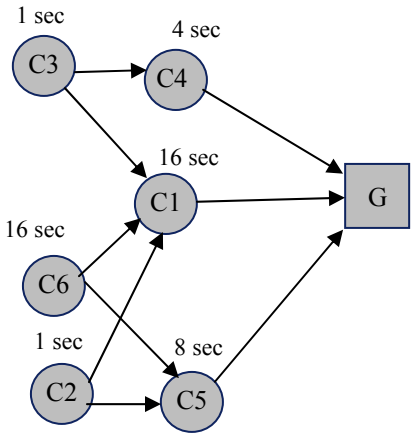
One Superframe - Length Based on Slowest scan rate – Duplicate slots would need to be scheduled in the Superframe for measurements that are scanned at a faster rate e.g. sixteen times for 1 second measurement in this example. Such duplication will increase the memory required for the Superframe.

To avoid any conflict between the slots reserved for process measurement and network management, the length of the network management Superframes may be configured to be an integer multiple of the fastest scan rate and configured to used slots that are not required for process measurement transmission. In this manner, slots may be allocated for the transmission of measurement without any conflict with the slots dedicated for management communications.

C.5. Scheduling Example – Multiple Hop

The previous example presented the ideal case where all network devices were connected through the Gateway through a single hop. In many cases this will not be the case – multiple hops will be utilized.

A WirelessHART Network is formed through the join process. When a Network Device does not communicate directly with the Gateway, then added communication slots must be reserved in the schedule for packet routing. These routing packets should be scheduled in the Superframe that corresponds to the downstream update rate - independent of the measurement scan period for the routing node. Also, if the network is configure to allow a network device to have multiple peers, then communication slots must be defined for each peer's data plus the messages it must route. When slots are reserved to transmit to both peers, the second slot in the Superframe is used for transmission only if no acknowledgement is received from its first peer in the Superframe. The impact of routing and for a network device to support multiple peers is illustrated through the following example – the example continues to follow the bioreactor example. In this example, the Network Devices are configured to support two peers and it is assumed that the following WirelessHART Network was formed as a result of the joining process. The network graph is shown below in Figure 57.



Device	Peer
C3	C4, C1
C6	C1, C5
C2	C1, C5
C4	G

Figure 57. Bio-Reactor Example – Multiple Hops

The resulting schedule is shown below in Figure 58.

Frame 0 – 1 Second Update

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5		TS99
0	C2=>C1	C2=>C5	C4=>G	C5=>G	C1=>G	C1=>G		
1	C3=>C4	C3=>C1						

Frame 1 – 4 Second Update

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5			TS399
2		C4=>G							

Frame 2 – 8 Second Update

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6		TS799
2	C5=>G								

Frame 3 – 16 Second Update

Ch Offset	TS0	TS1	TS2	TS3	TS4	TS5	TS6	TS7				TS1599
1			C6=>C5	C6=>C1			C1=>G	C5=>G				

Slots for Routing

Figure 58. Use of multiple frames to support different update rates

C.6. Updating Schedule for New Devices

The schedule for a wireless communications is gradually created as additional devices join the network. As was illustrated in the previous multi-hop example, the addition of a new device will required that the schedule be updated to allow the originating device to transmit the measurement and also for the value to be forwarded to the Gateway. If the scheduling approach outlined in the multi-hop example is followed, then the changes in the Graph to reserve dedicated slots for their communications will only impact the frames established for the new device and for the devices that are designed to route the message.

C.7. Actuator Setpoint – Regulating Valves

When a Network Device is an actuator that is a regulating valve, then the Host may periodically update the actuator setpoint as part of a batch or continuous control strategy. If the actuator is a blocking valve, then the setpoint will be updated on a change-of-state. In either case it is ideal to send the setpoint transmissions from the Gateway to Network Device associated with the actuator using *Graph routing*. One Superframe will created for each actuator to allocate communication slots for the associated transmissions based on a configured setpoint update period of a regulating actuator or the response time for setpoint changes to a blocking valve. The Superframe(s) defined for these downstream transmissions will be similar to the network management transmission Superframes except that there is only one target device and multiple paths are defined to reach this device. Thus, the Superframe length will be established by the configured setpoint update period. The Superframes created for actuator update will only contain slots needed for the host to direct packets to the devices that are actuators. In most applications the actuators will be a small percentage of the total number of network devices.

The previous batch bio-reactor example may be used to illustrate the superframe required for transmission of regulating valve setpoints. In this example, there are five regulating valves. The setpoint of these valves would be updated by the Host on a one second basis. If we assume the following WirelessHART Network is formed as a result of the joining process, then the Superframe for adjustment of the A1 actuator setpoints could be illustrated as shown below in Figure 59.

Table 20. Regulating Valves

Device	Measurement	Update Rate
A1	Feed Flow – FV201	1 sec
A2	Reagent flow – FV203	1 sec
A3	Coolant Flow – FV206	1 sec
A4	Vent Flow – FV208	1 sec
A5	Air flow – FV202	1 sec

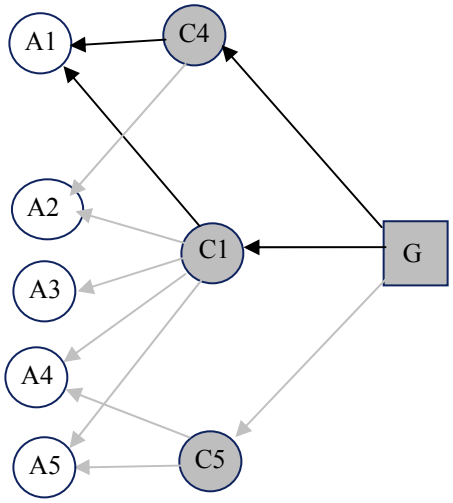


Figure 59. Bio-reactor Example Graph – Regulating Valve Setpoint

If slots were created for one re-try, then the setpoint update superframe would appear as shown Figure 60. Since the slots allocated in the Superframe are not used for process data and alerts, then there will be no conflicts in the schedule.

Setpoint Update Frame

Ch Offset	TS0	TS1			TS9	TS10	TS11	TS12	TS13	TS14	TS99
0					G=>C4	G=>C1					
1						C4=>A1	C1=>A1				
2											
3											

Figure 60. Bio-reactor Example Graph – Actuator Setpoint Update

When multiple paths are added in the schedule, then it is possible to support one or more re-tries using an alternate path within the same transmission cycle.

C.8. Actuator Setpoint – Blocking Valves

When a blocking valve is part of a network, then the setpoint will normally only be written to the device when the discrete setpoint changes. The communication of this new setpoint should be scheduled to be communicated from the Gateway to the blocking valve with minimum delay. However, unlike the setpoint updates that are sent to a regulating value, the communications to a

blocking valve must be a confirmed service to insure that the change in setpoint is not missed. Thus, extra communication slots must be reserved for the confirmation message.

The schedule required for the communications associated with the blocking valve is dependent on knowing the maximum response time that is allowed for a change in setpoint to be communicated to the blocking valve. For example, if the maximum response time is 2 seconds, then communication slots must be scheduled on a more frequent basis. For example, if the communication slots are reserved for 1 second, then depending on when the change arrives at the gateway it may be as long as 1 second before communication of the new setpoint value can be initiated. The total response time is the sum of this wait time plus and the time required to transmit and forward the message to the device. Since communications are scheduled, then the response time will on average be approximately equal to ½ of the communications period.

The bio-reactor process contains three blocking valves. The average response time to a setpoint change is one second. For this example, we assume that these devices joined the network as shown below in Figure 61.

Table 21. Blocking Valves

Device	Blocking Valve	Ave Response Time
B1	Charge Flow – FZ211	1 sec
B2	Harvest Flow – FZ212	1 sec
B3	Harvest Flow – FZ213	1 sec

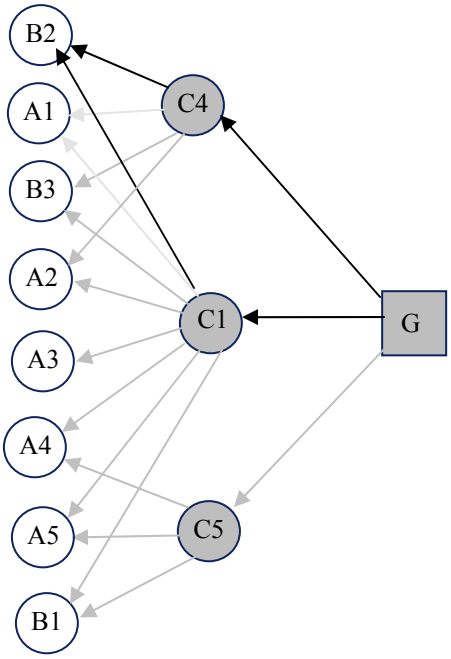


Figure 61. Bio-reactor Example Graph – Adding in Blocking Valves

The setpoint transmissions will be sent from the Gateway to the blocking valve using ***Graph routing***. A transmit superframe will created for each blocking valve. The superframe length will be set based on a configured response time for setpoint changes to a blocking valve. If slots were created for one re-try, then the setpoint transmit and response update superframe would appear as shown in Figure 62. Since the slots allocated in the superframe are not used for process data and alerts, then there will be no conflicts in the schedule.

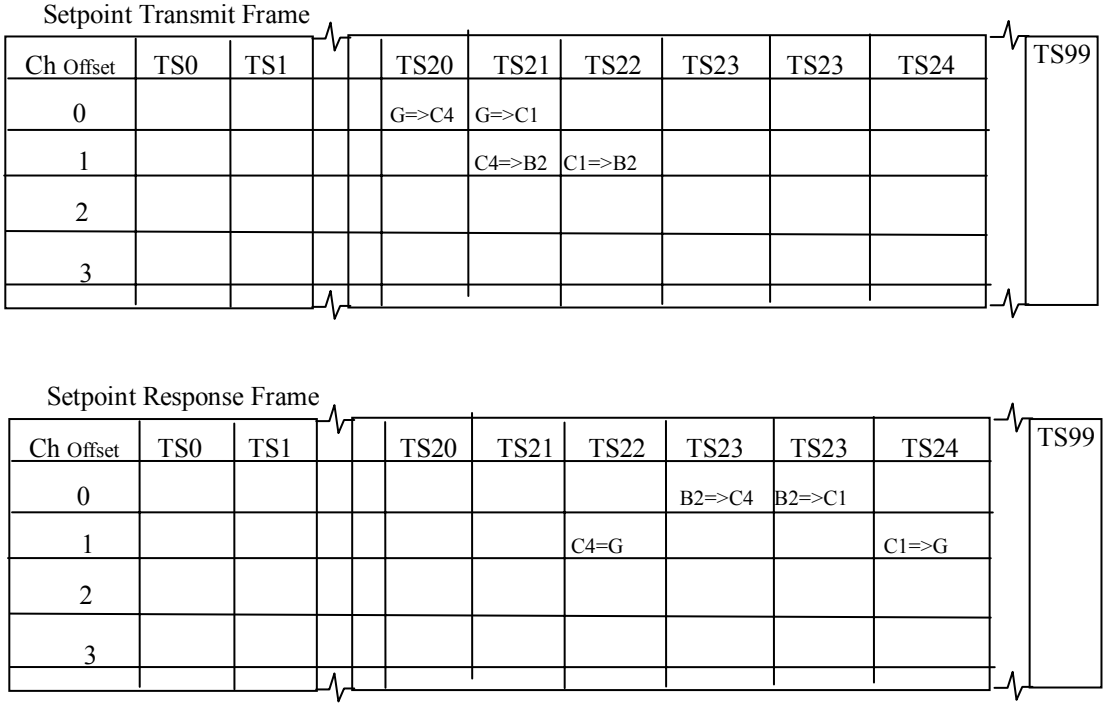


Figure 62. Bio-reactor Example Superframe – Adding in Blocking Valves

ANNEX D. REVISION HISTORY

D.1. Changes from Revision 1.0 to 1.1

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

D.2. Revision 1.0 (5 September 2007)

Updated based on Igls, Austria WG Comments