ENVISIONING • EMPOWERING • EXCELLING

MAXLINEAR

G.hn Spirit HN Software

User Guide

Revision History

Document No.	Release Date	Change Description
086UGR00	December 14, 2021	Initial release.

Table of Contents

List of Figures

List of Tables

Introduction

This document describes the main features included in the MaxLinear's Spirit HN software which is intended for home networking applications that use the G.hn 88LX5153A/88LX2741, 88LX5153/88LX2730, and 88LX5152/88LX2720 chipsets.

Firmware Description

The embedded microprocessor in the digital baseband processor (DBB) runs MaxLinear’s Spirit HN software. This firmware provides communication and management applications and a flexible API that you can use to build your own code. It also offers a high degree of customization of existing features. External processors are only required for specific applications. You can achieve the interconnection to external application processors by using SPI, MII/RGMII, SGMII, and UART interfaces.

The Spirit HN software manages the G.hn nodes intercommunication and is intended to support peer-to-peer connectivity in home networking scenarios.

The Spirit HN software is composed of several independent modules that you can add or remove depending on your needs.

Capabilities

The following table lists the capabilities supported by Spirit HN 7.12 GA.

Table 1: List of Capabilities Supported by Spirit HN 7.12 GA

Capability	Maximum Value
Number of supported nodes in the network	■ 14 nodes for PLC. ■ eight nodes for coax/phone (16 supported, only eight tested).
Number of connections per node	■ 13 connections (one connection per peer) for data and management in PLC. Seven in phone/coax (15 supported, only seven tested). ■ One connection for broadcast. ■ One connection-less port for registration.
Ethernet MAC addresses per G.hn domain	1024
Forward error correction (FEC) payload size	■ 120 bytes for broadcast connection. ■ 540 bytes for data and management.
FEC rates	1/2, 2/3, 5/6, 16/18, and 20/21
Number of simultaneous multicast channels routed	128

Firmware Core Features

This section describes the common core features included in the Spirit HN software.

Real Time Operating System and POSIX Interface Core

The embedded firmware runs on top of a customized uCOS-II kernel, a general-purpose real-time operating system (RTOS) that achieves low latency and great stability.

It provides a preemptive and a real-time deterministic multitasking environment that includes features such as interrupt handling, semaphores, event flags, message mailboxes, and queues, tasks, time, and timer management.

The RTOS services are not directly exposed to the software development kit (SDK) developer. Instead, a POSIX-compliant layer sits on top of the RTOS that helps you to migrate the existing applications.

Flash File System

The Spirit HN software implements a flash file system to manage the flash memory space that stores the firmware images, configuration files, and other kind of files required for upper applications such as log capture and web server.

The file system supports long names (up to 128 characters), directories, and concurrent access and also provides a fail-safe (FS) access to the flash memory. FS means that it is robust in case of power cuts or resets during write operations in flash. There are two copies of the file descriptor table, so that the consistency of the file system is preserved when a power interruption appears and files cannot be corrupted in case of a problem during write operations.

The file system also implements a wear leveling algorithm to equalize the use of the sectors and prevent early failure of a frequently accessed one. In addition, the file system simplifies access to the flash memory from the API by offering POSIX-standard functions for file manipulation.

Finally, the file system allows you to set read-only permissions on any file.

TCP/IP Stack

The Spirit HN software includes a TCP/IPv4 and IPv6 dual stack—in particular the Unicoi's TCPI/P stack stable version 8.11.0—that supports the following protocols: IP, UDP, TCP, ARP, and ICMP.

The stack itself is not needed for basic G.hn transceiver functionalities, but it is intended for remote accessibility, mainly for operation, maintenance, and configuration purposes. The UDP/TCP protocols allow you to create sockets with other IP machines and use high-level protocols such as FTP and HTTP.

The TCP/IP stack supports IP fragmentation.

The following table lists the main generic CFL parameters to configure the TCP/IP stack.

Table 2: TCP/IP Generic Configuration Parameters

Parameter	Description
`TCPIP.IPV4.IP_ADDRESS`	IPv4 address of the node.
`TCPIP.IPV4.IP_NETMASK`	IPv4 netmask of the node.
`TCPIP.IPV4.GATEWAY`	IPv4 gateway to connect the node to other LAN segments.
`TCPIP.IPV6.ENABLE`	■ TRUE: The IP stack is launched with IPv4 and IPv6 support. ■ FALSE: The IP stack is launched only with IPv4 support. This parameter takes effect only after a reset.
`TCPIP.IPV6.IP_ADDRESS`	IPv6 address of the node.
`TCPIP.IPV6.GATEWAY`	IPv6 gateway to connect the node to other LAN segments. **Note:** In IPv6, routes are automatically learned. Only set this parameter under exceptional circumstances (for example, routers do not send *router advertisement* packets) and in these cases ALWAYS use the link-local address of the router.

Trivial File Transfer Protocol Client

The Spirit HN software includes a Trivial File Transfer Protocol (TFTP) client that allows you to transfer files from/to a TFTP server. You can mainly use it to download new firmware versions into the nodes.

Supported file name length for transferred files can be up to 256 characters.

Table 3: Main Configuration Parameters Related to TFTP

Parameter	Description
`FLUPGRADE.GENERAL.HOST`	Server hostname or IP (IPv4 or IPv6) address. It applies only to Trivial File Transfer Protocol (TFTP) and File Transfer Protocol (FTP). The default port for the FTP server is 21. However, a different port can be indicated using the following notations: ■ `hostname:port` ■ `ip:port` For example, `10.10.1.1:210`. **Note:** The port cannot be indicated if the protocol is TFTP.
`FLUPGRADE.GENERAL.SOURCE`	Protocol used to transfer the file: ■ TFTP: Connects to a standard TFTP server. The server address is indicated by `FLUPGRADE.GENERAL.HOST`, port 69. ■ FTP: Connects to a standard FTP server. The server address and port are indicated by `FLUPGRADE.GENERAL.HOST`. ■ L2: MaxLinear proprietary Ethernet-level protocol.
`FLUPGRADE.GENERAL.STATUS`	Status message of the upload process. The possible values are: ■ *Ready: Initial status*. ■ *Ready: Finished correctly*. ■ *Failed*, plus a short description of the error.

File Transfer Protocol Client

The Spirit HN software includes a File Transfer Protocol (FTP) client that allows you to transfer files from/to an FTP server. You can mainly use it to download new firmware versions into the nodes.

Supported file name length for transferred files can be up to 256 characters.

Table 4: Main Configuration Parameters Related to FTP

Parameter	Description
`FLUPGRADE.GENERAL.HOST`	Server hostname or IP (IPv4 or IPv6) address. It applies only to TFTP and FTP. The default port for the FTP server is 21. However, a different port can be indicated using the following notations: ■ `hostname:port` ■ `ip:port` For example, `10.10.1.1:210`. **Note:** The port cannot be indicated if the protocol is TFTP.
`FLUPGRADE.GENERAL.SOURCE`	Protocol used to transfer the file: ■ TFTP: Connects to a standard TFTP server. The server address is indicated by `FLUPGRADE.GENERAL.HOST`, port 69. ■ FTP: Connects to a standard FTP server. The server address and port are indicated by `FLUPGRADE.GENERAL.HOST`. ■ L2: MaxLinear proprietary Ethernet-level protocol.
`FLUPGRADE.GENERAL.STATUS`	Status message of the upload process. The possible values are: ■ *Ready: Initial status*. ■ *Ready: Finished correctly*. ■ *Failed*, plus a short description of the error.
`FLUPGRADE.GENERAL.FTP_LOGIN`	FTP user login. It applies only to the FTP.
`FLUPGRADE.GENERAL.FTP_PASSWORD`	FTP user password. It applies only to the FTP.

Domain Name System Client

The Spirit HN software includes a Domain Name System (DNS) client into the firmware. DNS clients are also known as resolvers in the DNS protocol terminology.

This feature enables the translation of domain names into IP addresses and vice-versa.

This feature enables the configuration of remote servers using their URLs, such as a Network Time Protocol (NTP) server.

Table 5: DNS Configuration Parameters

Parameter	Description
`DNS.GENERAL.IPV4`	Domain name server IP (IPv4).
`DNS.GENERAL.IPV6`	Domain name server IP (IPv6).
`DNS.GENERAL.IPV4_TYPE`	Method used to assign the IPv4 DNS server address.
`DNS.GENERAL.IPV6_TYPE`	Method used to assign the IPv6 DNS server address.

Network Time Protocol Client

The Spirit HN software includes a Network Time Protocol (NTP) client into the firmware. The NTP is a protocol that synchronizes the clocks of computer systems over packet-switched, variable-latency data network. It uses the UDP port 123 as its transport layer and it is specially designed to resist the effects of variable latency (jitter).

This protocol enables the use of a real-time clock (RTC) in the system.

Table 6: NTP Configuration Parameters

Parameter	Description
`NTP.GENERAL.HOST`	URL or IP (IPv4 or IPv6) of the NTP server.
`NTP.GENERAL.HOST2`	URL or IP (IPv4 or IPv6) of the NTP server (2nd option).
`NTP.GENERAL.HOST3`	URL or IP (IPv4 or IPv6) of the NTP server (3rd option).
`NTP.GENERAL.HOST4`	URL or IP (IPv4 or IPv6) of the NTP server (4th option).
`NTP.GENERAL.HOST5`	URL or IP (IPv4 or IPv6) of the NTP server (5th option).
`NTP.GENERAL.ENABLED`	Enables and disables the NTP client.
`NTP.GENERAL.RESYNC_TIME`	Configures the resynchronization interval time (minutes).
`NTP.GENERAL.STATUS`	NTP client status. ■ Amount of disabled clients. ■ Unsynchronized clients: The absolute time is not set yet. ■ Synchronized clients: The device acquired accurate absolute time. ■ `Error_FailedToSynchronize`: The device failed to acquire accurate absolute time.

Dynamic Host Configuration Protocol Client

The Spirit HN software includes a Dynamic Host Configuration Protocol (DHCP) client to automatically configure the basic IP parameters (IP address, subnet, mask, default gateway address, option-82, option-125) from a DHCP server in the network.

Table 7: DHCP Configuration Parameters

Parameter	Description
DHCP.GENERAL.ENABLED_IPV4	Enables or disables the DHCP IPv4 client. It takes effect on the next reboot. Both IPv4 and IPv6 ENABLED parameters are independent and can be enabled at the same time.
DHCP.GENERAL.ENABLED_IPV6	Enables or disables the DHCP IPv6 client. Make sure that the IPv6 stack is also enabled. Otherwise, this parameter does not take effect at that time but takes effect on the next reboot. Both IPV4 and IPV6 ENABLED parameters are independent and can be enabled at the same time.
DHCP.GENERAL.LEASE_TIME_IPV4	Maximum lease time granted by the DHCP server for the current DHCP IPv4 address.
DHCP.GENERAL.LEASE_TIME_IPV6	Maximum lease time granted by the DHCP server for the current DHCP IPv6 address.
DHCP.GENERAL.SERVER_IPV4	IPv4 address of the current DHCP server.
DHCP.GENERAL.DUID_IPV6	You can use the DHCP Unique Identifier (DUID) to obtain an IP address from a DHCPv6 server.

HTTP Server

The Spirit HN software includes a HTTP server that implements a subset of RFC 2616 (HTTP/1.1) and provides an easy-to-use API around common HTTP items such as headers and request parameters. Its architecture is optimized for embedded systems and can run efficiently on limited-resource systems.

Configurable Flash Support

The Spirit HN software includes drivers for several flash devices. You can select different flash devices and develop your own flash device drivers based on the already available ones.

For more information about flash support, refer to the *G.hn Spirit Firmware Customization Programming Guide* (006PG).

Support for Loader

The Spirit HN software implements a loader module which supervises the initialization of the digital baseband processor (DBP) and any external component.

The loader prepares the required hardware blocks to gain access to DDR memory and the other interfaces, locating the firmware image in the flash file system, reading and unzipping it from flash into the RAM, and starting the execution of the firmware after the device initialization is completed.

There are two slots in the flash to store two images of the loader. It also enables a remote upgrade of the loader in a secure way (FS).

System Boot Process

The first code executed in the HN equipment is the ROM boot code. It configures the Ethernet interfaces and other basic hardware modules to access the external flash memory. Then it searches into the flash for a valid binary (the loader) to load into the memory.

The DBP reads the loader from the flash into the SRAM memory and the loader is executed. It also performs complex configurations such as clocks and external DDR memory. The loader can access the flash file system and locate the main firmware file, read it from the flash, unzip it, and load it into the suitable memory sections.

After the operation is completed, the main firmware code starts executing.

If the boot process fails at some point and cannot recover from it, the device enters in recovery mode. For more information, see "Recovery Mode".

Recovery Mode

If the flash memory is empty or corrupted, or the HN node is powered on while pushing the **CONFIG** button, the DBP boots through its Ethernet port and sends *ROM Boot* broadcast packets and waits for a binary file (to be loaded in memory) through the Ethernet interface following the Ethernet protocol. This way, you can load a firmware in the equipment and then recover the modem doing a flash upgrade from the Spirit Configuration Tool (SCT).

For more information, refer to the *Spirit Configuration Tool (SCT) User Guide* (052UG).

Flash Production Section

This is a section in the flash device to store sensitive configuration that is written in production. It is write-protected to prevent from undesired modifications.

The typical parameters stored in this section are:

- Device manufacturer.
- Device name.
- Device description.
- Device serial number.
- Device Ethernet MAC address.
- AFE calibration information.

To set these parameters, you can use the product configuration kit (PCK) tool provided along with the firmware release when generating a complete flash image and along with the production test kit (PTK) tool. For more information, refer to the help embedded in both tools delivered with the firmware.

Secure Upgrade

The secure upgrade (FS) of any flash binary section or file included in a Spirit firmware release is possible with backup images during the upgrade procedure of the:

- Firmware.
- Loader.
- Configuration files.
- Factory reset configuration.

To upgrade it, you can use the Layer 3 (FTP, TFTP) or Layer 2 (L2Upgrade).

> **Note:** MaxLinear recommends that you use L2Upgrade only for HN nodes connected locally through Ethernet.

You can also use the upgrade to update any other files included in flash FS. It is intended for specific applications such as the web server customization.

Table 8: Configuration Parameters Related to Main Upgrade

Parameter	Description
`FLUPGRADE.GENERAL.HOST`	Server hostname or IP (IPv4 or IPv6) address. It applies only to TFTP and FTP. The default port for the FTP server is 21. However, a different port can be indicated using the following notations: ■ `hostname:port` ■ `ip:port` For example, `10.10.1.1:210`. **Note:** The port cannot be indicated if the protocol is TFTP.
`FLUPGRADE.GENERAL.SOURCE`	Protocol used to transfer the file: ■ TFTP: Connects to a standard TFTP server. The server address is indicated by `FLUPGRADE.GENERAL.HOST`, port 69. ■ FTP: Connects to a standard FTP server. The server address and port are indicated by `FLUPGRADE.GENERAL.HOST`. ■ L2: MaxLinear proprietary Ethernet-level protocol.
`FLUPGRADE.GENERAL.STATUS`	Status message of the upload process. The possible values are: ■ *Ready: Initial status*. ■ *Ready: Finished correctly*. ■ *Failed*, plus a short description of the error.
`FLUPGRADE.GENERAL.SECTION`	Type of content of the file to upload: ■ Firmware, Loader, Config and Factory are specific system sections. ■ The File section uploads any file. ■ The Flash section upgrades the entire system. All sections are overwritten. ■ The Params_update section updates the value of a group of the configuration layer parameters. ■ The OSUP section performs the one step upgrade.
`FLUPGRADE.GENERAL.START`	When a value is written, this parameter starts the file upload. When a value is read, this parameter reports the status of the file transfer. The possible values are: ■ 1: In progress. ■ 0: Finished or not started.
`FLUPGRADE.GENERAL.STATUS`	Status message of the upload process. The possible values are: ■ *Ready: Initial status*. ■ *Ready: Finished correctly*. ■ *Failed*, plus a short description of the error.

Factory Reset

The factory reset operation:

- Recovers the configuration stored in the factory reset configuration file and applies it to the modems. You can customize the configuration file with the PCK that was defined in production, although you can also upgrade it remotely.
- Overwrites any settings that can be done afterwards. This includes the modem IP address.
- Can contain any set of parameters, from one single parameter to the complete parameter set. This file does not include the parameters stored in the production section.
- Can be initiated by using the SCT or by pressing the **CONFIG** button for more than 10 seconds (you can customize both the GPIO button and the time).

For more information about the factory reset functionality, see “Factory Reset Parameters” on page 77.

Ethernet Driver

The Spirit HN software allows the Ethernet device to connect to the DBB through the RGMII or the SGMII interface.

You can define the Ethernet interface type and its parameters (mode, IFG, delays, and MIIM mode) by using the `ETHIFDRIVER` parameters. Take into account the following limitations:

- The RGMII interface can be connected only to the ETHA port.
- The SGMII interface can be connected only to the ETHB port.
- You cannot use two interfaces at maximum speed simultaneously.

Table 9: Main Ethernet Driver Configuration Parameters

Parameter	Description
`ETHIFDRIVER.ETHA.ENABLED`	Specifies if the Ethernet interface A is enabled (YES) or disabled (NO).
`ETHIFDRIVER.ETHA.IFACE_TYPE`	Type of bus used in the Ethernet A interface. If the internal PHY device is connected to the Ethernet A interface, the type of bus used is SSMII. Otherwise, the options are MII, RGMII, or SGMII.
`ETHIFDRIVER.ETHB.ENABLED`	Specifies if the Ethernet interface B is enabled (YES) or disabled (NO).
`ETHIFDRIVER.ETHB.IFACE_TYPE`	Type of bus used in the Ethernet B interface. Currently, only the SSMII type is allowed. Other options are reserved.

You can configure the Ethernet link capabilities, duplex option, and check the status of the link by using the `ETHPHYCONF` parameters. Take into account the following considerations:

- If auto-negotiation is disabled (not recommended), the configured speed can be modified by using the `ETHPHYCONF` parameters.
- To force a specific speed and duplex configuration, MaxLinear recommends that you use auto-negotiation only enabling the desired capability. Disable other capabilities. Half-duplex is not supported.

Table 10: Main Ethernet PHY Configuration Parameters

Parameter	Description
ETHPHYCONF.ETHA.AUTONEG	Enables or disables the Ethernet A interface auto-negotiation. **Important:** When changing the AUTONEG value stored in flash, you MUST also configure both the SPEED and DUPLEX values stored in flash.
ETHPHYCONF.ETHA.LINK	Status of the Ethernet A interface link.
ETHPHYCONF.ETHA.SPEED	Speed of the Ethernet A interface. To modify it, disable the auto-negotiation in runtime. If the auto-negotiation is enabled, this parameter reports the auto-negotiation result and cannot be written.
ETHPHYCONF.ETHB.AUTONEG	Enables or disables the Ethernet B interface auto-negotiation. **Important:** When changing the AUTONEG value stored in flash, you MUST also configure both the SPEED and DUPLEX values stored in flash.
ETHPHYCONF.ETHB.LINK	Status of the Ethernet B interface link.
ETHPHYCONF.ETHB.SPEED	Speed of the Ethernet B interface. To modify it, disable the auto-negotiation in runtime. If the auto-negotiation is enabled, this parameter reports the auto-negotiation result and cannot be written.
ETHPHYCONF.HGF.ETH_IF_TYPE	Associated with the LCMP HGF parameter used to obtain the maximum supported type (100/1000) of the main Ethernet interface.

Watchdog Module

The Spirit HN software implements a watchdog module to monitor the correct operation of the firmware. It continuously checks there is no memory corruption, the RTOS operates properly, and there is no problem in any functional module in the adapter.

In case the watchdog module found something wrong, you can configure it to reset the device, storing information about the error before the reset to be checked afterwards.

G.hn PHY

This section describes the G.hn physical layer implemented in the Spirit HN software.

PHY Layer Support

The Spirit HN software supports the following physical layer G.hn specifications:

- *ITU-T G.9960 Standard:* Defines the core of the physical layer for G.hn systems.
- *ITU-T G.9964 Standard:* States the requirements of the power spectral density (PSD) that G.hn system needs to meet.

The following subsections provide more details about the level of compliance.

- Physical medium attachment sub-layer support as described in the *ITU-T G.9960 Standard (sub clause 7.1.3)*.
- Physical medium dependent sub-layer support for PLC mode as described in the *ITU-T G.9960 Standard (sub clause 7.1.4)*.
- Predefined and runtime bit allocation table (BAT) support as described in the *ITU-T G.9960 Standard (sub clause 7.1.4.2.2)*.
- Predefined and runtime bit and Tx port mapping allocation table (BMAT) support as described in the *ITU-T G.9963 Standard (sub clause 7.1.4.4.3)*.
- Automatic selection of robust communication mode (RCM) transmission mode as described in the *ITU-T G.9960 Standard (sub clause 7.1.3.3)*.
- PHY frame formats as defined in the *ITU-T G 9963 Standard (sub clause 7.1.2.1)*.

Profiles Support

The following table lists the G.hn PHY profiles defined in the Spirit HN software for PLC, coax and phone wirings.

Table 11: PHY Profiles Supported in Spirit HN

G.hn Profile	Running PHY ID	Notes
PLC 100Mhz SISO	23	■ Uses phase-neutral or two wires. ■ Frequency range: 2–80MHz ■ Carriers: 3201 ■ Tsym: 40.96us (without CP) ■ PSD max level [0–30MHz]: –57dBm/Hz ■ PSD max level [30–80MHz]: –72dBm/Hz
PLC 100Mhz MIMO	7	■ Uses phase-neutral-ground (for three-wires injection) or four wires. ■ Frequency range: 2–80MHz ■ Carriers: 6402 ■ Tsym: 40.96us (without CP) ■ PSD max level [0–30MHz]: –57dBm/Hz ■ PSD max level [30–80 MHz]: –72dBm/Hz
Coax 200Mhz SISO	25	■ Frequency range: 2–190MHz ■ Carriers: 962 ■ Tsym: 5.12us (without CP) ■ PSD max level [2–5MHz]: Slope from –94dBm/Hz to –76dBm/Hz ■ PSD max level [5–190MHz]: –76dBm/Hz
Phone 200Mhz SISO	26	■ Frequency range: 2–187MHz ■ Carriers: 1894 ■ Tsym: 20.48us (without CP) ■ PSD max level: –76dBm/Hz

The Spirit HN PLC PHY modes can interoperate between 2x2 MIMO devices and 1x1 devices. They can maximize the performance by using MIMO processing techniques based on the *ITU-T G.9963 Standard*.

The following figures show the PSD levels for the PHY profiles supported in the Spirit HN software.

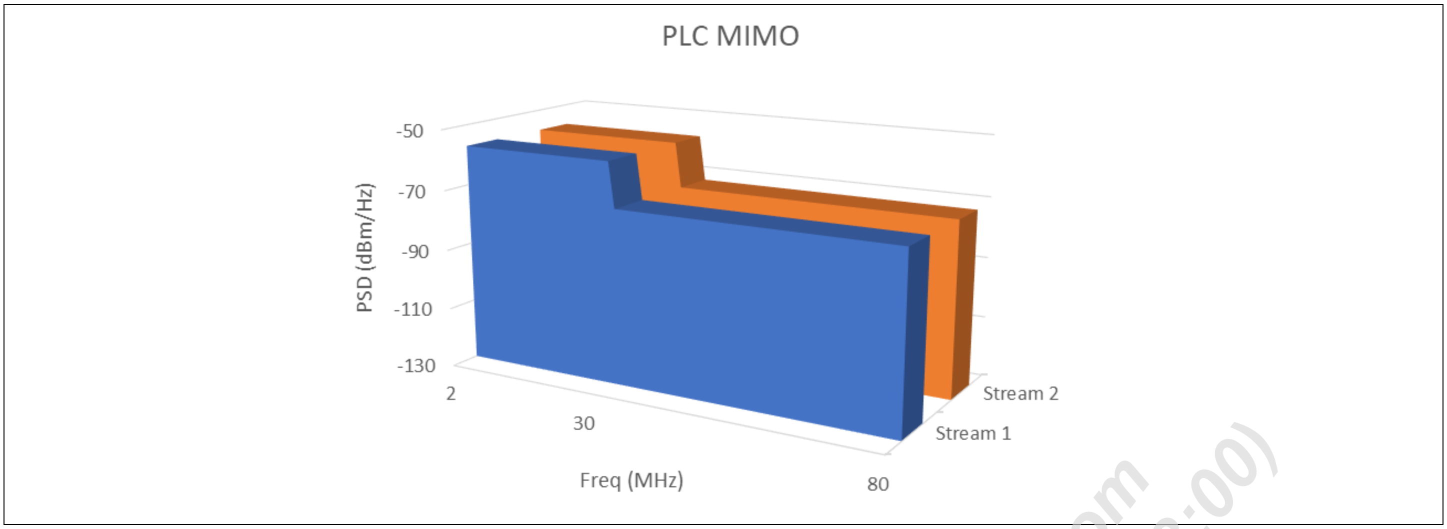


Figure 1: PSD Levels Description for MIMO PLC Supported in Spirit HN

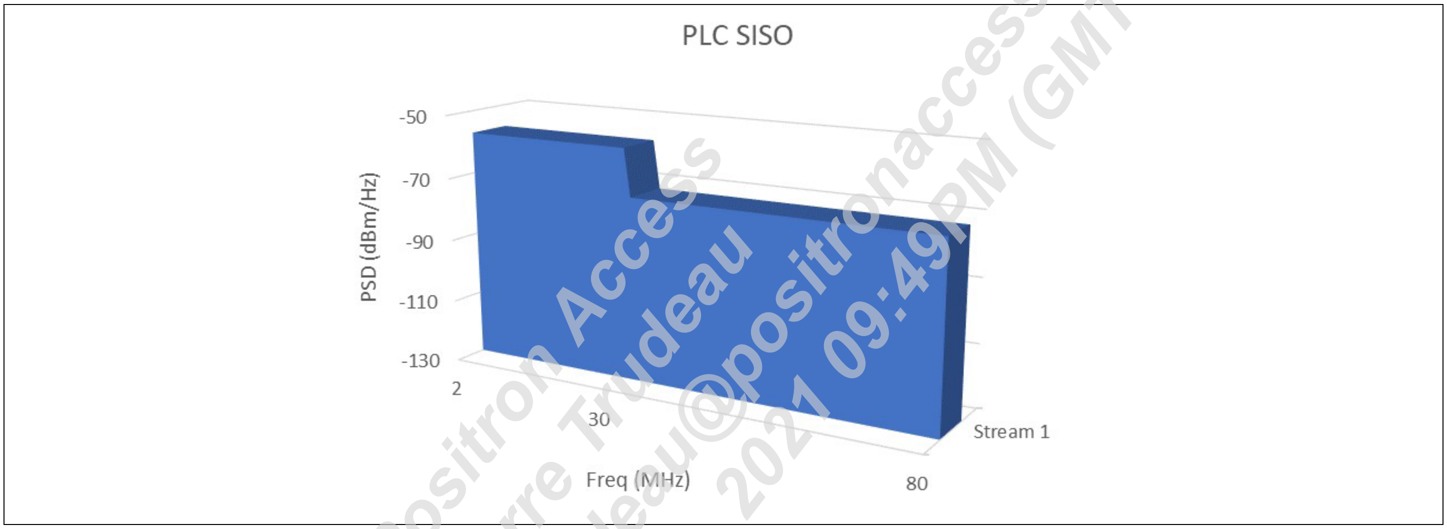


Figure 2: PSD Levels Description for SISO PLC Supported in Spirit HN

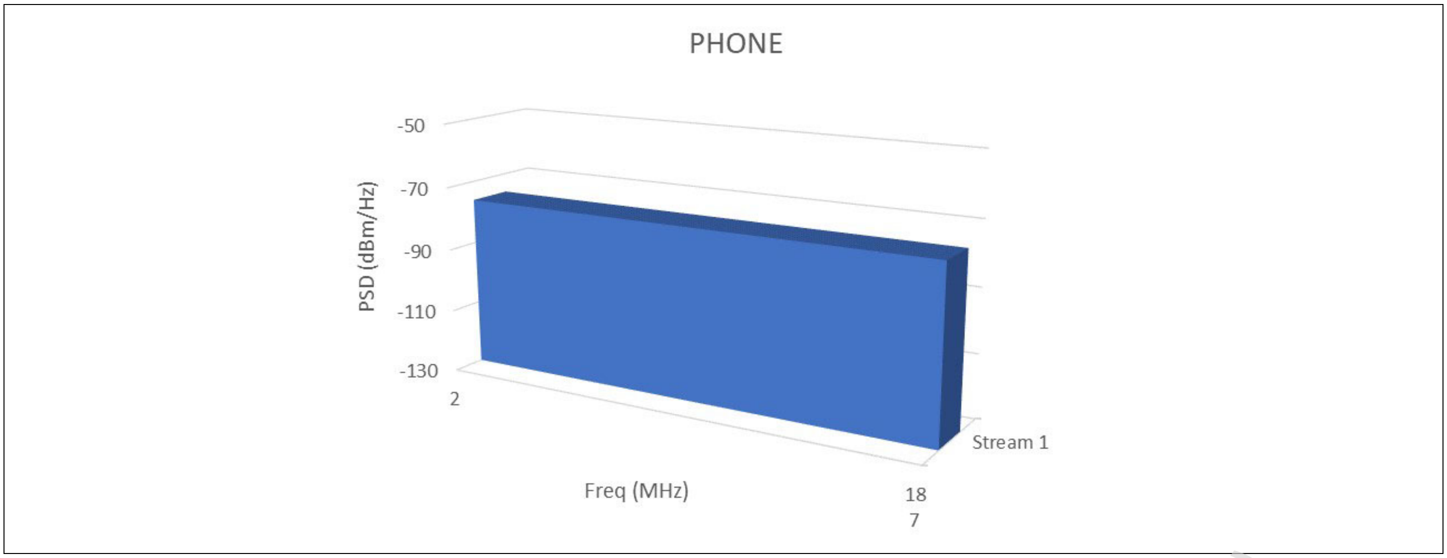


Figure 3: PSD Levels Description for Phone Supported in Spirit HN

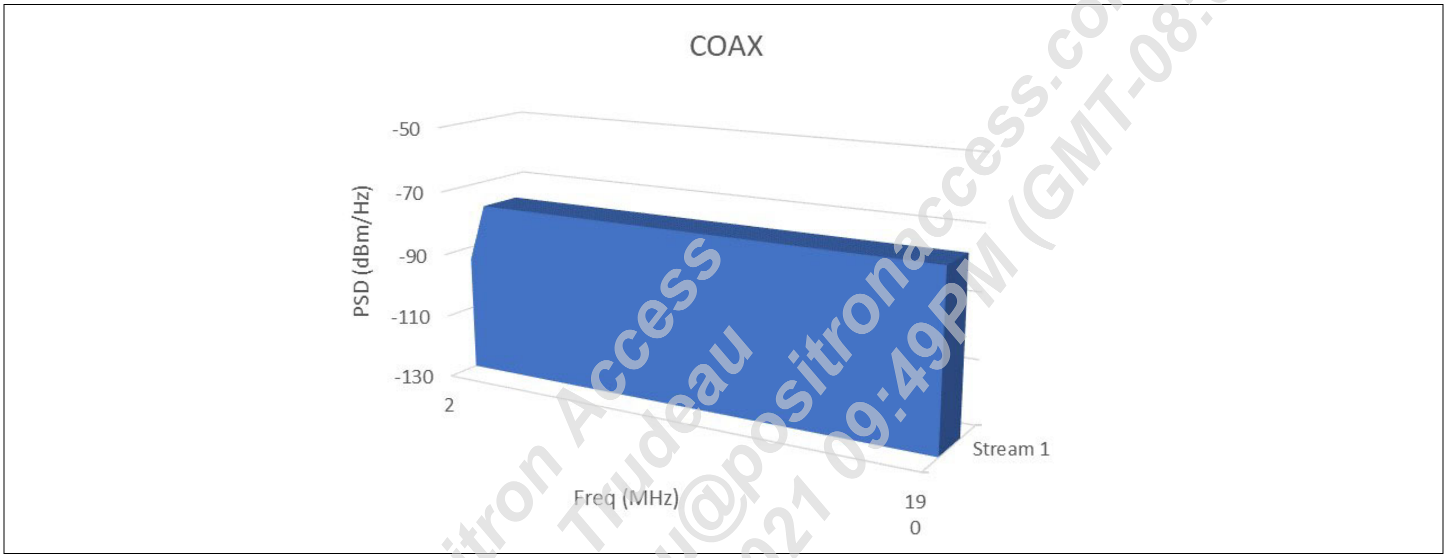


Figure 4: PSD Levels Description for Coax Supported in Spirit HN

CENELEC EN 50561-1 Support

The Spirit HN software is compliant with the *CENELEC EN 50561-1* for PLC media only (phone and coax are not affected by this recommendation).

This software enables the certification of PLC products to be commercialized in the European Union.

The *EN 50561-1* is a harmonized standard under the EMC directive of the European Commission. Compliance with the *EN 50561-1* gives a presumption of compliance with part of the EMC requirements stated in the EMC directive. Compliance with the EMC requirements is mandatory for CE marking.

The software includes:

- Addition of fixed aeronautical notches.
- Limitation of the radiation in absence of data.
- Dynamic PSD control based on attenuation.
- Notching of frequencies used by the radio broadcast service. They can be fixed or dynamic based on the detection if the modem is able to detect. In this case, it is a dynamic notching based on detection.

The following table lists the parameters required to configure the modem to be *EN 50561-1* compliant.

Table 12: Configuration for CENELEC EN 50561-1 Compliance

Parameter	Value
`POWERMASK.GENERAL.50561_1_NDEPTH_EN`	YES
`POWERMASK.CENELEC.NOTCHES_ENABLE`	YES
`LINKADAPTATION.MONITORS.DYN_PSDC_ENABLE`	YES
`INTERFMITIGATION.DYNOTCH.ENABLE`	YES
`POWERSAVING.GENERAL.MODE`	1 (Ethernet link)
`POWERSAVING.GENERAL.IDLE_TIME`	6

The following table lists the main *CENELEC EN 50561-1* configuration parameters.

Table 13: Main CENELEC EN 50561-1 Configuration Parameters

Parameter	Description
`LINKADAPTATION.MONITORS.DYN_PSDC_ENABLE`	Enables or disables the dynamic PSD ceiling performed by the Rx.
`LINKADAPTATION.MONITORS.DYN_PSDC_IQRTHR`	Configures the dynamic PSD ceiling IQRTHR parameter. IQR is an expression of the flatness of the channel frequency response (CFR).
`LINKADAPTATION.MONITORS.DYN_PSDC_IQR`	Shows the last IQR calculus performed over the CFR measure. This value must be lower than the threshold. If a problem occurs (`DYN_PSDC_STATUS` is not 0x7F), review your design or alternatively increase the IQRTHR.
`LINKADAPTATION.MONITORS.DYN_PSDC_ATT`	Shows the last attenuation calculus performed over the CFR measure.
`LINKADAPTATION.MONITORS.DYN_PSDC_REQ_VAL`	Shows the last PSD ceiling value requested to the Tx side (see "Dynamic PSD" on page 18).
`LINKADAPTATION.MONITORS.DYN_PSDC_STATUS`	Shows the current status of the PSDC monitor. The 0x7F value means everything is correct and runs, otherwise there is an internal error code.
`INTERFMITIGATION.GENERAL.NBAND_NOISE_THR`	Powers the threshold to consider the background noise as a disturbing additive white Gaussian noise (AWGN). If a problem occurs (`DYN_PSDC_STATUS` is not 0x7F), review your design or alternatively increment this THR.
`INTERFMITIGATION.GENERAL.AWGN_NOISE_THR`	Powers the threshold to consider a detected signal as a disturbing narrow-band noise. If a problem occurs (`DYN_PSDC_STATUS` is not 0x7F), review your design or alternatively increment this THR.

PSD and Notching Configuration

The Spirit HN software allows you to define the notching configuration and the PSD.

In the G.hn software, the powermask is built after processing the following four settings in the following order:

1. Notches (`CFL POWERMASK.USER`, `POWERMASK.VENDOR`, `POWERMASK.REGULATION.NOTCHES` (see Table 14).
2. Calibration (`CFL POWERMASK.CALIBRATION`).
3. Shape (`CFL POWERMASK.SHAPEDEF`).
4. Dynamic (`CFL POWERMASK.DYNAMIC`). Similar to Shape but designed to change the powermask *on the fly* for coexistence with other technologies (G.fast, DSL).

Notches

- Vendor powermask: The product manufacturer can enable or disable the notches by using the PCK. For more information, refer to the help embedded in the PCK tool delivered along with the firmware.
- User powermask: The customer can configure the notches by using any of the provided configuration methods, such as the SCT. For more information, refer to the help embedded in the SCT tool delivered along with the firmware.
- Regulation powermask: The injected PSD meets with the limits defined in the *EN 50561-1 Standard*. The following table lists the default notches defined.

Table 14: Regulation Notches

Notch type	Freq Start (KHz)	Freq End (KHz)
IARU	1800	2000
IARU	3500	4000
IARU	7000	7300
IARU	10100	10150
IARU	14000	14350
IARU	18068	18168
IARU	21000	21450
IARU	24890	24990
IARU	28000	29700
Aeronautical	2850	3025
Aeronautical	3400	3500
Aeronautical	4630	4700
Aeronautical	5250	5450
Aeronautical	5480	5680
Aeronautical	6525	6685
Aeronautical	8815	8965
Aeronautical	10005	10100
Aeronautical	11275	11400
Aeronautical	13260	13360
Aeronautical	17900	17970
Aeronautical	21924	22000
Aeronautical	26960	27410

Note: The IARUs are notches required for PLC and phone in FCC and CENELEC recommendations. Aeronautical notches are required for PLC and CENELEC recommendations.

The following table lists the parameters required to enable or disable the different sets of notches.

Table 15: Parameters to Enable/Disable Different Sets of Notches

Parameter	Description
`POWERMASK.VENDOR.NOTCHES_ENABLE`	Enables the vendor-defined set of notches. This parameter cannot be changed in runtime, only using the PCK or SDK.
`POWERMASK.USER.NOTCHES_ENABLE`	Indicates if the user notches are applied when the `POWERMASK.GENERAL.NOTCHES_UPDATE` parameter is executed.
`POWERMASK.REGULATION.NOTCHES_ENABLE`	Enables the user-defined set of notches. This parameter can be changed in runtime. The notch configuration is applied when the `POWERMASK.GENERAL.NOTCHES_UPDATE` parameter is written.
`POWERMASK.CENELEC.NOTCHES_ENABLE`	Indicates if the CENELEC notches are applied (adding aeronautical notches) when the `POWERMASK.GENERAL.NOTCHES_UPDATE` parameter is executed. Only for PLC.

Calibration

Calibration powermask is intended to correct possible PSD ripples produced by analog components, typically to produce a flat PSD transmission performing pre-equalization.

You can ignore this set of parameters and use `POWERMASK.SHAPEDEF` instead.

Shape Definition

The Spirit HN software allows you to configure independently the powermask (PSD) for each MIMO channel.

If necessary, you can do it using the configuration layer parameter listed in the following table.

Table 16: Shape Definition Values Configuration Parameter

Parameter	Description
`POWERMASK.SHAPEDEF.PM_PORTS[6][33]`	Indicates the port to apply the defined PSD shape. In MIMO, there are two Tx ports and the PSD can be defined independently. ■ 0: All (port 1 and port 2). ■ 1: Port 1 only (phase-neutral). ■ 2: Port 2 only (phase-neutral-ground).

Example:

```
POWERMASK.SHAPEDEF.FREQS.1.0 = 45000,54000,70000,75000,80000,1990,2000,3750,5190,5200,
8500,8510,11990,12000,17700,17710,35000,35010,45000,54000,60000,65000,70000,75000,80000,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

POWERMASK.SHAPEDEF.PM_VALS.1.0 = 0,-8,-25,-28,-29,0,-40,-40,0,-48,-48,-24,-24,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0

POWERMASK.SHAPEDEF.PM_PORTS.1.0 = 1,1,1,1,1,2,2,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0

POWERMASK.SHAPEDEF.MODE = 7,0,0,0,0,0
```

The following table lists the values that correspond to the settings used in the shape definition example.

Table 17: Values Assigned in Shape Definition Example

Frequency	Attenuation	Port
45000	0	1
54000	–8	1
70000	–25	1
75000	–28	1
80000	–29	1
1990	0	2
2000	–40	2
3750	–40	2
5190	0	2
5200	–48	2
8500	–48	2
8510	–24	2
11990	–24	2

Dynamic

The Spirit HN software includes a new set of parameters added to the configuration layer which allows you to change the powermask *on the fly* (POWERMASK.DYNAMIC).

The POWERMASK.DYNAMIC set of parameters is applied to the powermask after applying notches (regulatory, vendor, and user) and SHAPEDEF, and the resulting powermask is the combination of all the notches, taking the most restrictive value for each.

Powermask Definition

Customers can define up to six sets of masks. Each mask is defined by the following three parameters (same configuration as POWERMASK.SHAPEDEF):

- POWERMASK.DYNAMIC.FREQS.X.0: It indicates the breakpoint frequency of the PSD internal descriptor. The size of the array is 128 (KHz).
- POWERMASK.DYNAMIC.PM_VALS.X.0: It indicates the breakpoint gains of the PSD internal descriptor. The size of the array is 128 (Unit: 0.25dB). Use only negative—attenuation—values. As it occurs with POWERMASK.SHAPEDEF.PM_VALS, the values are relative, not absolute.
- POWERMASK.DYNAMIC.PM_PORTS.X.0: It indicates the port to apply the defined PSD dynamic. In MIMO, there are two Tx ports and the PSD can be defined independently. The size of the array is 128. The possible values are:
  - 0: Both ports.
  - 1: Port 1
  - 2: Port 2 (MIMO only).

Powermask Setting

To select one of the six powermasks to configure, use the `POWERMASK.DYNAMIC.RUNNING` parameter. It stores the index of the dynamic array set to apply the values from 0 to 6. 0 is not a valid value (no dynamic powermask is applied). Only 1–6 are valid values.

Example: If `POWERMASK.DYNAMIC.RUNNING` = 3,

```
POWERMASK.DYNAMIC.FREQS.3.0
```

```
POWERMASK.DYNAMIC.PM_VALS.3.0
```

```
POWERMASK.DYNAMIC.PM_PORTS.3.0
```

The change is non-disruptive, which means that the modem does not unregister from the network, and it is applied immediately.

`POWERMASK.DYNAMIC.LAST_VALID` is a parameter where the latest valid index value can be stored, so it can be read and applied at bootup. It can be useful if the modem is rebooted during a powermask operation transition.

Example: `POWERMASK.DYNAMIC.RUNNING = 1`

```
POWERMASK.DYNAMIC.FREQS.1.0 = 5000,7500,7501,11999,12000,30000,5000,7500,7501,11999,12000,
30000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
POWERMASK.DYNAMIC.PM_VALS.1.0 = -12,-12,0,0,-12,-12,-36,-36,0,0,-36,-36,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
POWERMASK.DYNAMIC.PM_PORTS.1.0 = 1,1,1,1,1,1,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
```

The following figures show the PSD applying the dynamic powermask of the example.

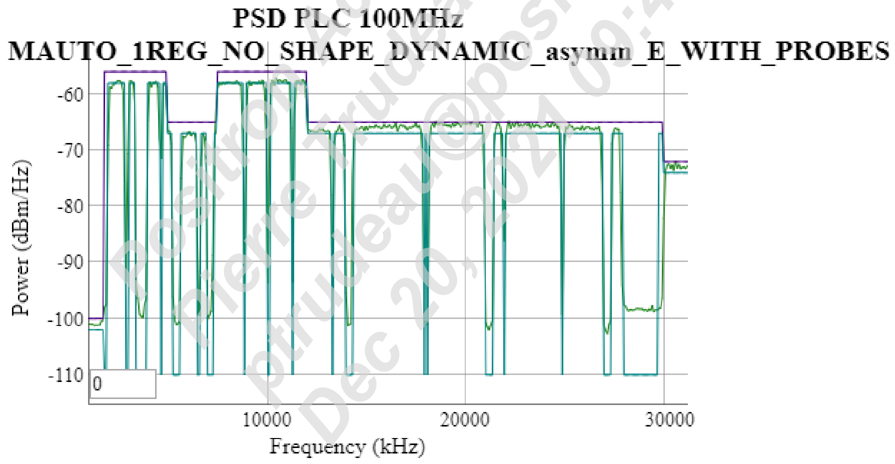


Figure 5: PSD After Applying Dynamic Powermask (CH1)

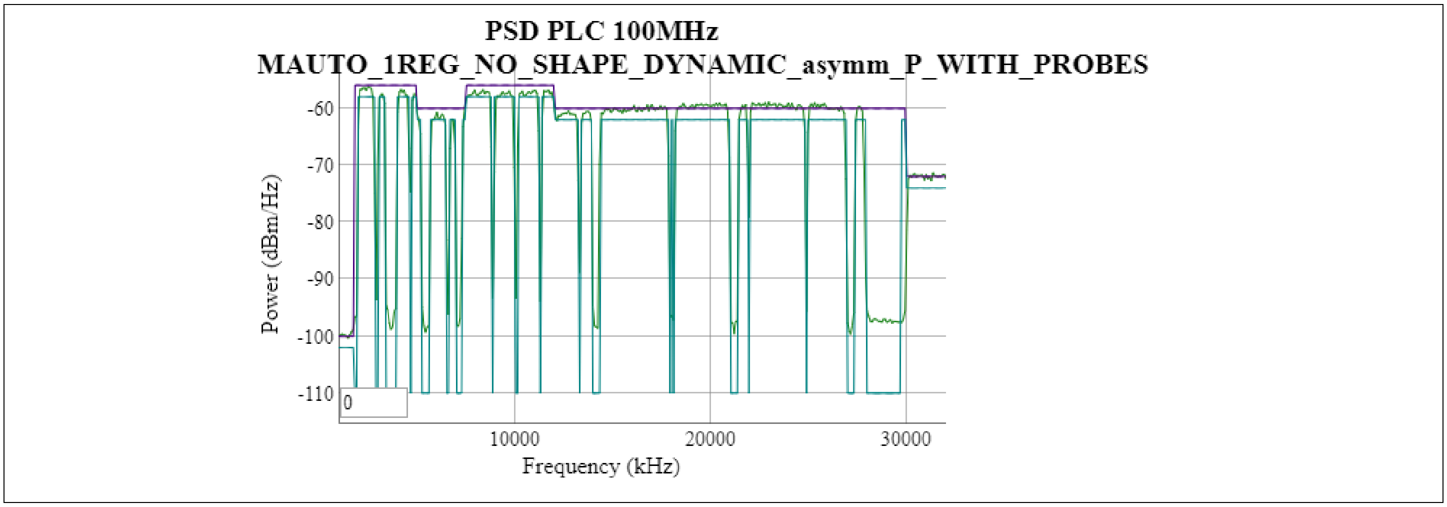


Figure 6: PSD After Applying Dynamic Powermask (CH2)

Dynamic PSD

The Spirit HN software supports the automatic control of the injected PSD.

The receiver node decides the PSD to inject. Based on the attenuation estimation and the signal-to-noise ratio (SNR) of the link, it decides the PSD that maximizes the bit loading.

This control of the PSD is calculated for each pair of nodes that have direct communication. Depending on the receiver node, the PSD injected can be different on each transmission.

The dynamic PSD applies only to data and PROBE PHY frames, while the PHY frames management, such as MAP or ACKs, is always transmitted using the maximum PSD.

The PSD changes from –56dBm/Hz to –96dBm/Hz in five steps. The following table lists the dynamic PSD steps.

Table 18: Dynamic PSD Steps

Dynamic PSD Index	Low band (2–30MHz)	High band (≥30MHz)
PSDC Idx 3	–56dBm/Hz	–72dBm/Hz
PSDC Idx 5	–60dBm/Hz	–72dBm/Hz
PSDC Idx 13	–76dBm/Hz	–76dBm/Hz
PSDC Idx 18[1]	–86dBm/Hz	–86dBm/Hz
PSDC Idx 23[1]	–96dBm/Hz	–96dBm/Hz

[1] idx means index.

Interference Generator Mode

The Spirit HN software allows you to select MAC behavior when there is no data to transmit. You can select one of the following modes:

- Normal mode: MAC remains silent when there is nothing to transmit (default mode).
- Force Tx mode: MAC transmits PROBE frames, which means that it forces PROBE transmission when there is a transmission opportunity and no data nor management.
- Force Silent mode: MAC limits transmission to management traffic only, thus minimizing the presence of the node in the channel.

Table 19: General Parameter Related to Interference Generator Mode

Parameter	Description
PHYMNG.GENERAL.TX_MAC_MODE	Configures the behavior at the MAC layer for transmissions. The possible values are: ■ 0: Default behavior. The node transmits whenever it has data or management to transmit. ■ 1: PROBEs are transmitted when there is no data to transmit. ■ 2: Silent mode. Only management frames can be transmitted.

The Force Tx mode measures interference in xDSL systems. By default, all nodes are configured in mode 0, but when it is necessary to measure interference, a round-robin is performed with each node. Thus only one modem at a time is configured and transmits PROBEs while the others remain in Force Silent mode.

This way, xDSL can estimate the interference that the G.hn PLC devices receive and using the POWERMASK.DYNAMIC mechanism can reduce the PSD in the most interfering devices.

MSPS and PHYMNG.GENERAL.TX_MAC_MODE Interaction

To configure PHYMNG.GENERAL.TX_MAC_MODE = 1, disable MSPS (MSPS.L1.ENABLE = NO or MSPS.L1.ACTIVE_SLOTS_FORCED = 16). If the modem enters in idle mode due to MSPS, few PROBEs are sent.

Waterfilling and PSD Limit (MIMO)

When configuring the PSD limits, take waterfilling into account:

- If certain carriers of a port (1 or 2) are nulled, the corresponding carriers of the other port (2 or 1) have a PSD increment of 3dB.
- To make sure that a PSD limit is not exceeded, consider having additional 3dB in MIMO.
- If an asymmetric powermask is configured (decreasing further in the port 2), it is very likely that the PSD increment occurs in the port 1.
- When measuring the PSD with default PROBEs, this effect is not observed because it uses a default bit and Tx port mapping allocation table (BMAT) without waterfilling.

Channel Adaptation

One of the main issues with G.hn is setting the correct bit loading for each link, especially in the PLC which is a difficult channel that suffers from noises, impedance changes, etc.

The following figure shows the four components of the channel adaptation process.

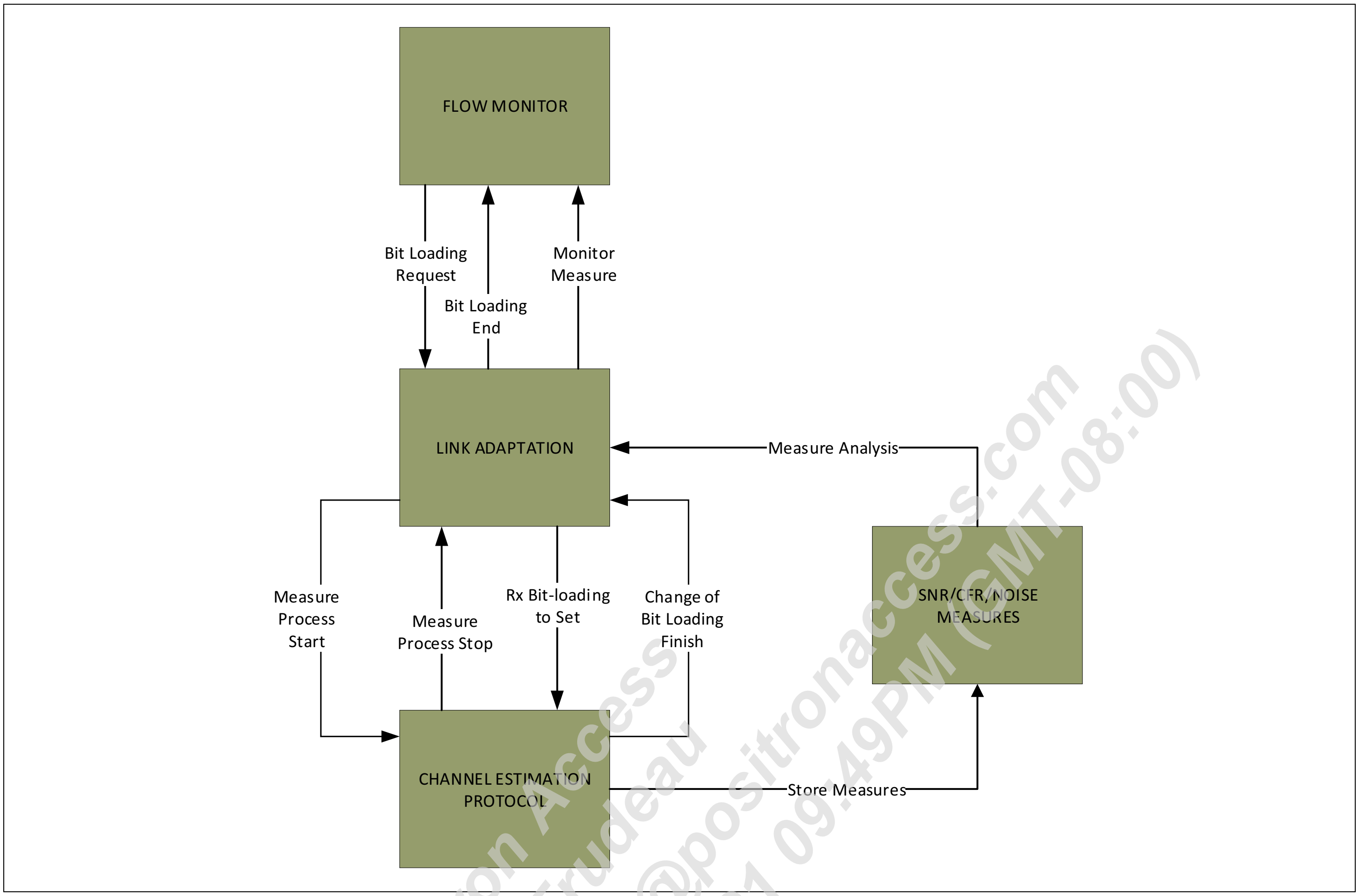


Figure 7: Channel Adaptation Components

The four components shown in Figure 7 are:

- Flow monitor—It retrieves information on FEC (forward error correction), MAC, and channel statistics per link (FEC errors, frame errors, attenuation, etc.). If the flow monitor decides that a bit loading re-adaptation is required, it requests the link adaptation to obtain a new bit loading.
- Link adaptation—It is responsible for the trigger measure process for channel estimation and it analyzes the measures obtained (SNR/CFR/noise) and decides:
  - Bit loading.
  - FEC rate.
  - PSD of transmitter.
  - Precoding (MIMO only).
  - Waterfilling (MIMO only).

  The link adaptation also informs the FM that the process is complete.

- Channel Estimation protocol (Chest)—It is responsible for:
  - Supporting the G.hn *ITU-T G. 9961 Standard* for requesting SNR probes with the desired settings.
  - Configuring the hardware to achieve the desired requested measure (SNR, CFR, background noise, etc.)
  - Storing the measures obtained in memory.
  - Informing the link adaptation that the measurement process is complete.
  - Receiving the bit loading decided by the link adaptation.
  - Negotiating the change of bit loading with the remote node (described in the G.hn *ITU-T G. 9961 Standard*).
  - Informing the link adaptation that the bit loading change protocol is complete.
  - Modifying—in Tx Chest—the bit loading on the request of the remote node which is an asynchronous event decided by the peer, not internally.
- Measure manager—The driver is responsible for:
  - Receiving the requests for measures.
  - Managing the priority between different requests and putting them in a queue.
  - Configuring the hardware block to obtain the measure.

A distinction must be made between phone/coax channel adaptation and PLC channel adaptation.

PLC Channel Adaptation

PLC channels are continuously subject to variations due to noise, attenuation, and impedance changes. Good and frequent channel measurement is essential to adapt to these changes and optimize the bit loading and throughput. This section covers the different aspects considered in the Spirit HN software to perform this optimization.

PLC Channel Description

PLC home channels are very demanding regarding bit loading changes to follow channel changes, since the powerline wire is not originally designed for data transmissions and the variety of devices plugged in.

There are different disturbances in the channel:

- Impulsive noise: It is typically produced by motors that are not synchronous with the electrical mains. It is unpredictable and has short duration (less than 100us).
  - Action: None. L2ACK takes care of the block error rate (BLER) and retransmits in a very efficient way.
- Background noise: Stable noise present in the channel increasing the noise floor.
  - Action: The SNR snapshot produces a good estimation of the channel, so that the bit loading can be adapted properly.
- Impedance changes: They are typically produced by chargers, led bulbs, etc. They are synchronous with the electrical mains. They are present in almost all PLC channels with an impact depending on the proximity of the disturbance source to the modem. The issue is not only the disturbance itself, but also the fact that the channel is different after the change.
  - Action: Set different bit loadings along the MAP cycle.

The following figure shows the SNR response at different stages of the MAP cycle.

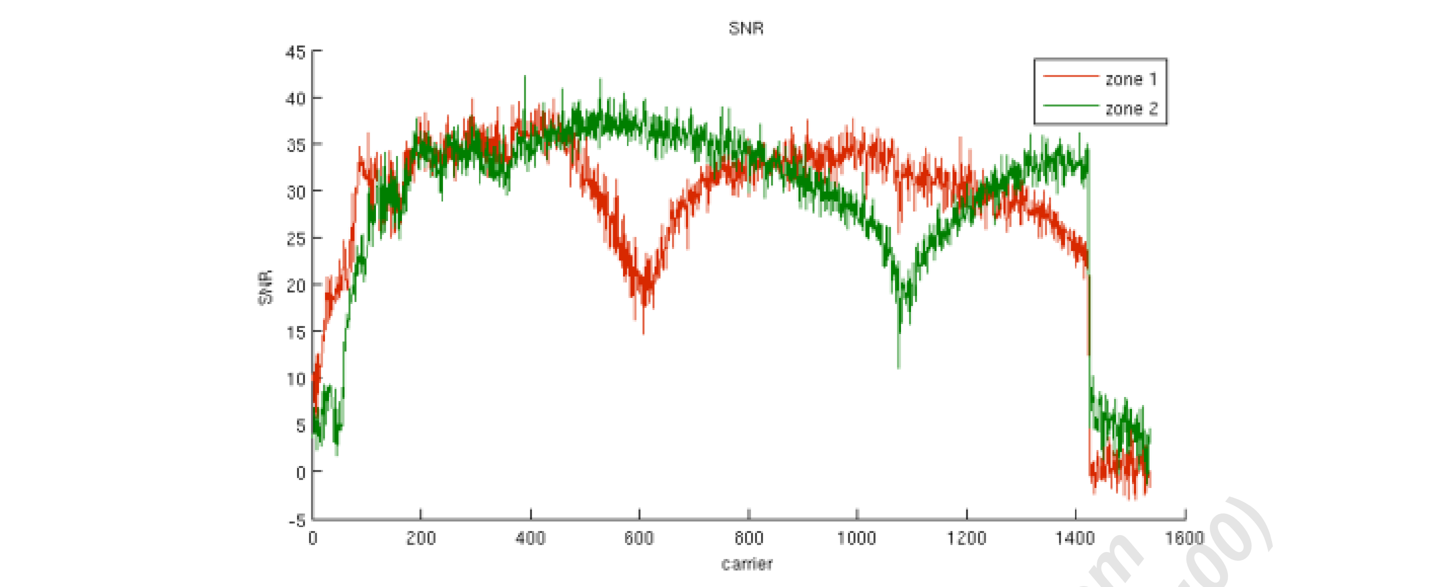


Figure 8: SNR Response at Different Stages of the Cycle

Bit Loading/Regions

In PLC, there are four tonemaps (or regions). Each tonemap is valid for 1/16 of the MAP cycle (2.5ms at 50Hz, 2.0833 at 60Hz) and is reused four times along the MAP cycle, as shown in the following figure.

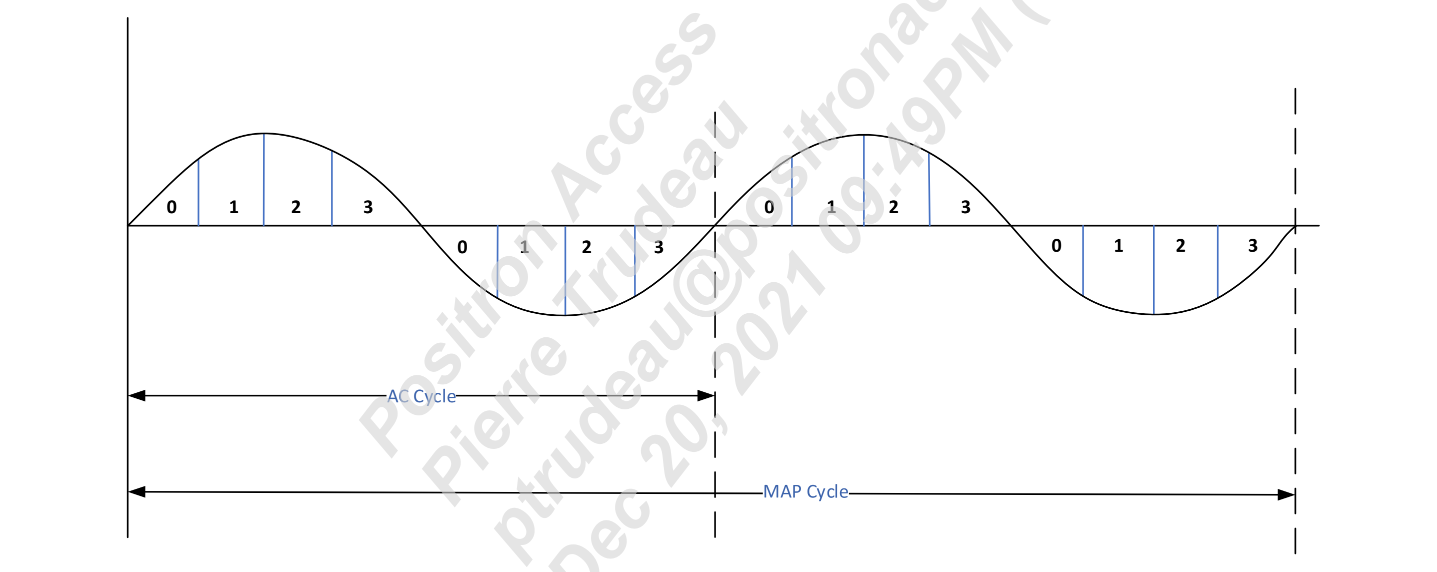


Figure 9: Region MAP

In addition to the bit loading, the receiver indicates to the transmitter:

- The selected FEC rate.
- The PSD level.

A fifth robust region is calculated as a conservative merge of all regions covering the whole MAP cycle. It is not used until a sudden change dramatically increasing the BLER is detected in the channel. Then, the bit loading of the fifth region is selected while a new adaptation process is triggered.

Precoding/Waterfilling

In the case of MIMO, the bit loading has two additional parameters:

- Precoding: A pair of angles is associated to each carrier to change the transmitted power of each carrier on each digital-to-analog converter (DAC). This technique is intended to reduce the crosstalk between the two streams, thus maximizing the SNR.
- Waterfilling: It can be better to have only one spatial stream with 2×P power than two spatial streams with one P power for each. For example, when the weakest spatial stream does not have enough SNR to achieve the minimum bits per carrier (BPC) scheme, it is particularly effective in channels with low SNR.

These two dimensions are part of the channel adaptation. For more information about MIMO specification, refer to the G.hn *ITU-T G.9963 Standard*.

Channel Adaptation Types

Depending on the severity of the BLER, the flow monitor triggers one type of adaptation or another. There are mainly two types of adaptation:

- Fast adaptation: The flow monitor observes a sudden and severe change in the BLER with a high level. It is intended to be a dramatic change in the channel so the current bit loading is no longer valid and therefore you must apply a new set of bit loading urgently.
  The main driver of this type of adaptation is speed, even if you do not define the optimum set of bit loading that the system needs to unblock the traffic reception because it is almost stalled.
- Soft adaptation: After a fast adaptation or when the flow monitor observes an abnormal BLER or a channel change in one or more regions, it triggers a soft adaptation.
  If it is based on the received data and can include one, two, three, or four regions, the main driver in that case is to achieve the optimum performance but not speed.
  In this type of adaptation, precoding and waterfilling are also calculated.

Phone/Coax Channel Adaptation

The phone and coax have the following characteristics for the channel adaptation in the Spirit HN software:

- Only one bit loading is used per link (one for the Rx and one for the Tx).
- It is only supported in SISO (although the hardware supports phone MIMO, it has not been developed for HN).
- No PSD control is needed.
- Only fast adaptation is supported, there is no need for soft adaptation.

Channel adaptation in devices that use coax or phone media is straightforward, as communication takes place in exclusive wires. In this case, the following assumptions can be made:

- The channel is considered stable.
- Changes in noise, interference, and impedance are minimum.
- There is no synchronous dependence on the electrical mains.

One SNR snapshot is sufficient to decide the bit loading for each link. Only SNR PROBEs (special frames defined in the G.hn *ITU-T G.9960 Standard*) are required to estimate the SNR.

The flow monitor continues to monitor the BLER and SNR estimation on data frames to decide if it is necessary to request a new bit loading.

Main Channel Adaptation CFL Parameters

These are the main configuration layer (CFL) parameters which provide information related to the channel estimation. The enable/disable parameters allow you to configure different flavors of the channel estimation, to meet the different requirements of each medium.

Table 20: Main Channel Estimation CFL Parameters

Parameter	Description
`LINKADAPTATION.MONITORS.DYN_PSDC_ENABLE`	Enables dynamic PSD control.
`LINKADAPTATION.PRECODING.ENABLE`	Enables or disables precoding (for debug purposes).
`LINKADAPTATION.WATERFILLING.ENABLE`	Enables or disables waterfilling (for debug purposes).
`CHEST.INFO.TX`	Information per link and per region in the Tx. ■ `SnID`: Session node ID (DID of the DST node). ■ `BPS[region]`: BPS for this SnID and this region (sum of BPCs). ■ `FECR[region]`: FEC rate for this SnID and this region. ■ `RCM[region]`: Indicates if the robust communication mode (RCM) is used in this SnID and this region. ■ `nREP[region]`: Number of repetitions used in this SnID and this region. Only when using the RCM.
`FLOWMONITOR.STATS.LINK_STATUS`	Information per link on the received frames, %ERROR

Interference Mitigation

G.hn devices can cause interference in other communication systems, either by radiation or by induction in other cables.

This component provides a framework to install different *interference mitigation clients* that can be used to reduce the effect of such interference. These clients include the code necessary to measure, evaluate, and mitigate potential interference caused to other non-G.hn devices. CENELEC has defined a set of procedures that are implemented in these clients.

The general approach is to try to detect the presence of devices that can potentially be affected by G.hn interference and reduce the PSD in a way that mitigates that. This usually implies measuring the presence of signals over a threshold in a certain band or set of bands.

You can achieve this reduction of PSD by notching or by making adjustments to the Tx PSD in a set of carriers or even by adjusting the overall Tx power level (ceiling).

> **Note:** The interference mitigation functionality is available in the Spirit HN SDK as source code, which allows customers to extend the functionality and to adapt it to the future regulation in terms of coexistence with other technologies.

Table 21: General Parameters Related to Interference Mitigation Functionality

Parameter	Description
`INTERFMITIGATION.GENERAL.ENABLED_AT_BOOT`	If set to *YES*, the interference mitigation functionality is enabled at bootup. The default value is YES.
`INTERFMITIGATION.GENERAL.ENABLED`	Runtime value for the interference mitigation general functionality. If set to *YES*, the functionality is enabled and runs.
`INTERFMITIGATION.GENERAL.NBAND_NOISE_BW`	Bandwidth of the narrow-band noise to search for. Units are in Hz.
`INTERFMITIGATION.GENERAL.NBAND_NOISE_THR`	Power threshold to consider a detected signal as a disturbing narrow-band noise. Units are in dBm/Hz.
`INTERFMITIGATION.GENERAL.AWGN_NOISE_THR`	Power threshold to consider the background noise as a disturbing additive white Gaussian noise (AWGN). Units are in dBm/Hz.
`INTERFMITIGATION.GENERAL.MEAS_TRIED`	Counter of processed channel measures (valid or not) including all types of measurements.

Each different interference mitigation client deals with a specific type of interference and they are implemented as self-contained blocks that can be installed into the interference mitigation framework. One or more of these clients can be installed and run at the same time.

Each client defines the measurements or detection procedure that it requires to detect the presence of other devices and the actuation procedure to use to reduce the PSD in a way that mitigates interference. This actuation uses the powermask component to modify the output power accordingly.

The following figure shows the different elements of the process. In this particular example, there are three different clients.

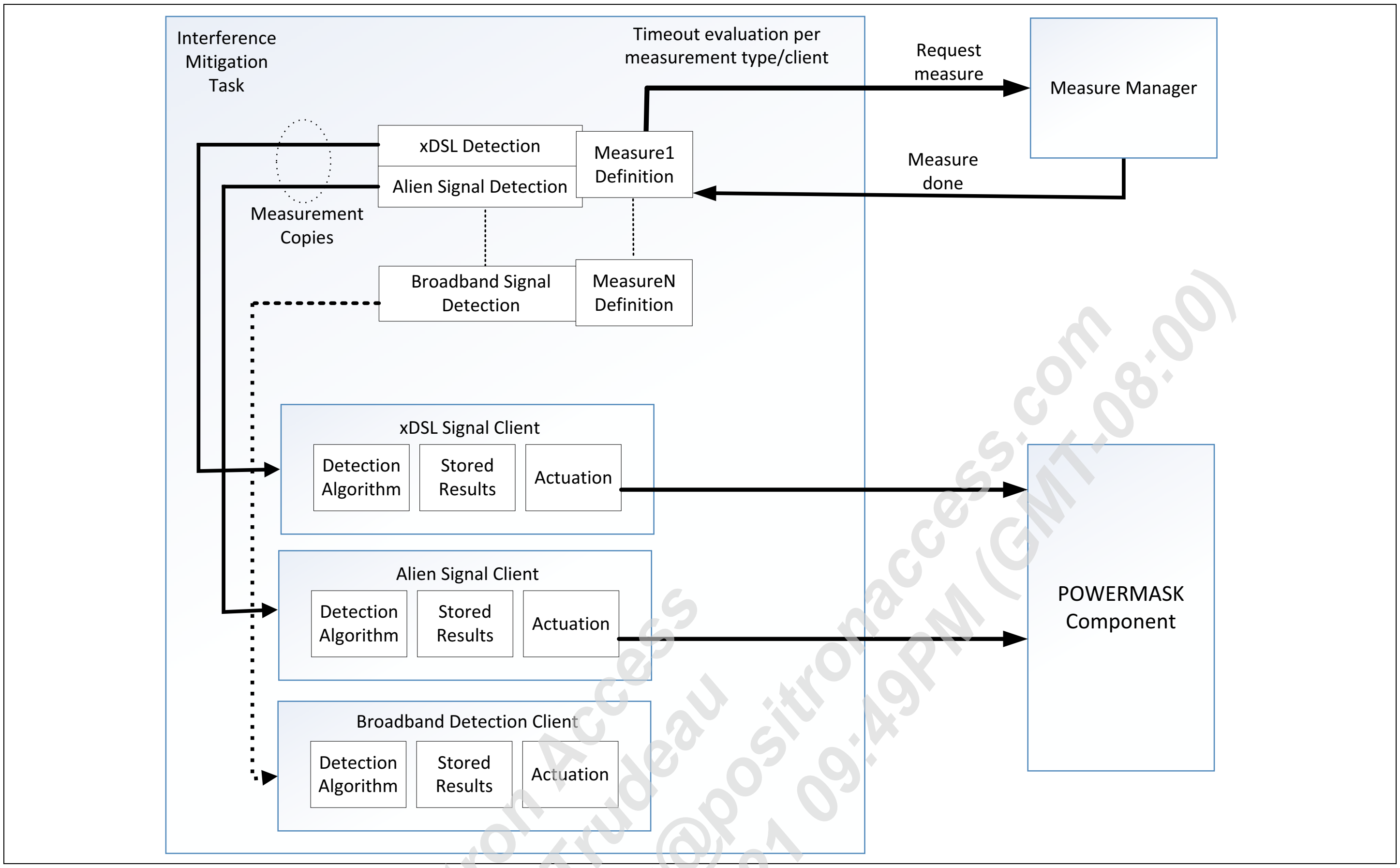


Figure 10: Interference Mitigation Process Basic Block Diagram

The interference mitigation component code is public and allows customers to implement their own interference mitigation clients, in addition to the ones already provided.

The clients already included are:

- Alien signal detection.
- Dynamic notching.
- xDSL.
- G.fast.

The following sections describe these clients.

Alien Signal Detection

It uses periodic background noise measurements to detect signals that are above a certain PSD threshold and have a minimum constant frequency width. The threshold and frequency band width of these signals can be configured to suit the detection of particular signals, such as DOCSIS. The output is a list of frequency bands where signals are detected.

Once a signal is detected in a band, you can perform an actuation to—for example—reduce the power in that band.

Table 22: Parameters Related to Interference Mitigation Alien Signal Detection Functionality

Parameter	Description
`INTERFMITIGATION.ALIENSIGNALDET.ENABLED`	Enables or disables the interference mitigation to alien networks functionality. ■ YES: Enabled. ■ NO: Disabled.
`INTERFMITIGATION.ALIENSIGNALDET.FREQ_TOL`	Frequency tolerance to assign a new detected band to an existing one.
`INTERFMITIGATION.ALIENSIGNALDET.MIN_BW`	Minimum detected bandwidth to consider the background noise as a possible alien signal.
`INTERFMITIGATION.ALIENSIGNALDET.N_BAND_DET`	Number of possible detected alien signals.
`INTERFMITIGATION.ALIENSIGNALDET.PERIOD_MS`	Period in msecs where a measurement is requested by the `AlienSignalDet` client.
`INTERFMITIGATION.ALIENSIGNALDET.THR`	Power threshold to consider the background noise as a possible alien signal.
`INTERFMITIGATION.ALIENSIGNALDET.BAND_DET`	Array of parameters that define a possible alien signal. ■ StartFreq: Frequency where the possible alien signal starts (in kHz). ■ StopFreq: Frequency where the possible alien signal ends (in kHz). ■ TimeStamp: Last time this channel is detected (in seconds since modem bootup).

Dynamic Notching (Part of CENELEC EN 50561-1)

The Spirit HN software is compliant with the *CENELEC EN50561-1*, including dynamic frequency exclusion and dynamic power control features for PLC media.

This software enables the certification of PLC products to be commercialized in the European Union.

The *EN 50561-1* is a harmonized standard under the EMC directive of the European Commission. Compliance with the *EN 50561-1* gives a presumption of compliance with part of the EMC requirements stated in the EMC directive. Compliance with the EMC requirements is mandatory for CE marking.

The software uses a periodic background noise measurement to detect one or more broadcast radio station tones that match a defined set of frequencies and power levels. Reducing interference for this set of radio frequencies is a requirement to meet the *CENELEC EN50561-1 Standard*.

The following table lists the different frequency ranges where PLC transmission must be removed when radio stations use them.

Table 23: Permanent or Dynamically Excluded Frequency Ranges

Excluded Frequency Range (MHz)	Service
2.30–2.498	Broadcasting
3.20–3.40	
3.90–4.05	
4.75–5.11	
5.75–6.20	
7.20–7.70	
9.30–9.95	
11.55–12.10	
13.55–13.90	
15.05–15.85	
17.40–17.90	
18.90–19.02	
21.45–21.85	
25.65–26.10	

Note: The bands in the table include frequency ranges allocated under Article 5 of the ITU Radio Regulations to the Broadcasting Service, plus a realistic appraisal of use for broadcasting under Article 4.4 of the ITU Radio Regulations.

When detecting each of the broadcast stations, the standard describes how the frequencies close to the broadcast signals are excluded from transmission by the local node, essentially applying a notch to the Tx power in the bands where the radio stations are detected.

Table 24: Parameter Related to Interference Mitigation Dynamic Notching Functionality

Parameter	Description
`INTERFMITIGATION.DYNOTCH.ENABLED`	Enables or disables the dynamic notching functionality. ■ YES: Enabled. ■ NO: Disabled.

xDSL

xDSL is a family of different profiles, each with different bands assigned for uplink and downlink. For now, only 998ADE17 profiles must be detected and protected.

These profiles use the bands listed in the following table.

Table 25: 998ADE17 and 998ADE35 Profile Bands

Profile 998ADE17					
DS Band	**Fstart (Mhz)**	**Fstop (Mhz)**	**US Band**	**Fstart (Mhz)**	**Fstop (Mhz)**
DS1	0.138	3.75	**US0**	-	-
DS2	5.2	8.5	**US1**	3.75	5.2
DS3	12	17.664	**US2**	8.5	12
Profile 998ADE35					
DS Band	**Fstart (Mhz)**	**Fstop (Mhz)**	**US Band**	**Fstart (Mhz)**	**Fstop (Mhz)**
DS1	0.138	3.75	**US0**	0.025	0.138
DS2	5.2	8.5	**US1**	3.75	5.2
DS3	12	35.328	**US2**	8.5	12

The detection of these profiles is based on one or more of the bands assigned to them. The US bands are typically detected with higher power than the DS bands, because the devices are physically closer. For this reason, detection uses one or more US bands. Additional criteria, such as sharp edges or profiles of the detected signals, can be added to improve detection of the DSL signals.

The actuation focuses on the DS bands, since they can be affected the most by interference because their level is much lower on the reception side.

Detection and actuation are limited to the range of frequencies used by G.hn.

Table 26: Parameters Related to Interference Mitigation xDSL Functionality

Parameter	Description
`INTERFMITIGATION.XDSL.ENABLED`	Enables or disables the xDSL interference mitigation functionality. ■ YES: Enabled. ■ NO: Disabled.
`INTERFMITIGATION.XDSL.BAND_DET`	Array of parameters that define a possible xDSL signal. Each row contains: ■ Freq: Frequency where the possible xDSL signal starts (in kHz). ■ Detected: Indicates if this frequency is currently detected. ■ TimeStamp: Last time this frequency was detected (in seconds).
`INTERFMITIGATION.XDSL.PRES_SEARCH_FREQ`	Each pair of values indicates the frequencies to search (maximum of 5): ■ StartFreq: kHz ■ StopFreq: kHz If the values are set to *0*, no probable DOCSIS channel is detected.
`INTERFMITIGATION.XDSL.DETECTED`	Indicates if a possible xDSL signal is detected. Complete this field when all detected information must be deleted.
`INTERFMITIGATION.XDSL.FREQ_TOL`	Frequency tolerance to assign a new detected band to an existing one (in kHz).
`INTERFMITIGATION.XDSL.THR`	Power threshold to consider the background noise as a possible xDSL signal (in dBm/Hz).
`INTERFMITIGATION.XDSL.TIMEOUT`	Timeout to remove the detected flag if the xDSL signal is no longer detected.
`INTERFMITIGATION.XDSL.PERIOD_MS`	Period in MS for the measurements associated to the xDSL client.

G.fast

G.fast is a TDD signal and the detection cannot only rely on identification of frequency-based patterns. Instead, certain timing information must be extracted from the measured background noise that matches the periods used in G.fast signals.

The DS and US signals are detected with different PSD levels. The duration of the DS and US transmissions can be detected from this difference in levels. The ratio between the DS and US durations must match the asymmetrical allocation of time channel for them.

The G.fast client is based on the CENELEC cognitive G.fast detection document.

Notes:

- G.fast detection has not been tested. It is provided as an example of implementation.
- The reduction of injected power is not yet defined by the *EN 50561-4 Standard*, so the Spirit HN software does not yet implement the reduction of power.

Table 27: Parameters Related to Interference Mitigation G.fast Functionality

Parameter	Description
`INTERFMITIGATION.GFAST.ENABLED`	Enables or disables the G.fast interference mitigation functionality. - YES: Enabled. - NO: Disabled.
`INTERFMITIGATION.GFAST.MIN_NUM_PEAKS_DETECTED`	Minimum number of detected peaks in a measurement to consider a signal present.
`INTERFMITIGATION.GFAST.MAX_NUM_PEAKS_DETECTED`	Maximum number of detected peaks in a measurement to consider a signal present.
`INTERFMITIGATION.GFAST.MIN_PEAK_WIDTH`	Number of consecutive samples higher than a threshold to consider a G.fast peak.
`INTERFMITIGATION.GFAST.PERIOD_MS`	Period in msecs when a measurement is requested by the G.fast client.
`INTERFMITIGATION.GFAST.SIGNAL_DETECTED`	Set to *YES* if a G.fast signal is detected in the current period.
`INTERFMITIGATION.GFAST.DETECTED_TIMESTAMP`	Timestamp in seconds of the last detection of a G.fast signal.
`INTERFMITIGATION.GFAST.NUM_ON_DETECTIONS`	Number of consecutive positive detections to set `SIGNAL_DETECTED` to *YES*.
`INTERFMITIGATION.GFAST.NUM_OFF_DETECTIONS`	Number of consecutive negative detections to set `SIGNAL_DETECTED` to *NO*.

G.hn MAC and QoS

The G.hn *ITU-T G.9961 Standard* describes the reference models and functionality for the data link layer components.

DBB-based devices that run the Spirit HN software support both domain master (DM) and end point (EP) roles.

The Spirit HN software supports most of the features described in the G.hn *ITU-T G.9961 Standard*. The following sections describe this support as well as additional features as recommended by the HomeGrid Forum (HGF).

Multinode Capabilities

The following table lists the maximum number of nodes supported in an HN domain, depending on the medium and the SISO/MIMO mode.

Table 28: Maximum Number of Nodes per Domain

PLC		Phone	Coax
SISO	**MIMO**	**SISO**	**SISO**
14	14	16 (only 8 tested)	16 (only 8 tested)

Spirit HN 7.12 does not support the use of MAPs, so full visibility is required between all EP nodes and the DM in the same domain.

EP nodes that have no visibility to each other are supported as long as they have visibility to the DM. For more information, see "Partial Visibility Support" on page 42.

Spirit HN Node Functions

In a G.hn network, there are two types of roles:

- Domain master (DM).
- End point (EP).

G.hn builds a self-organized network. This means that you do not need to declare a node as a DM of the system. The roles are assigned automatically. When a node becomes a DM, the rest of the nodes (EPs) register to it.

Any G.hn node implements the following features described in the *ITU-T G.9961 Standard (sub clause 8.5)*:

- Synchronization of the MAC cycle and synchronized transmissions: The EP synchronizes its MAC cycle and MAC clock with the DM and follows the transmission opportunity (TXOP) and time slots assignments in the MAC cycle.
- Routing of application data primitives (ADPs).
- Broadcast of the logical link control (LLC) frames.
- Retransmissions and Layer 2 acknowledgments.
- Network Admission Protocol (to request access to the domain).
- Domain master selection: A domain master selection protocol is used to dynamically select a single domain master in the presence of multiple nodes capable of operating as a domain master.

The domain master runs the following additional features described in the *ITU-T G.9961 Standard (sub clause 8.6)*:

- Network Admission Protocol (to grant access to the domain): It means granting the domain registration to new nodes.
- Synchronization with external source: When operating on an AC powerline medium provided by a public utility with a nominal cycle frequency of 50 hertz or 60 hertz, the domain master must synchronize the MAC cycle with the powerline cycle.
- Routing.
- Generate MAP: Management frame that publishes the medium access schedule at each MAC cycle.

Medium Access Control

The Spirit HN software implements a TDMA-based medium access control synchronized with the AC mains cycle (50 or 60Hz depending on geography) for PLC profiles. For SISO coax and phoneline profiles, the domain master generates an internal synchronization signal every 40 milliseconds.

For more information about the MAC sub-layer, refer to the *ITU-T G.9960 Standard.*

The Spirit HN software uses a contention-free transmission opportunity (CFTXOP) for MAP transmission and shared transmission opportunities (STXOPs) for data and management transmissions, as described in the *ITU-T G.9961 Standard (sub clause 8.3)*.

The Spirit HN software supports non-persistent transmissions, and the scheduling schema assigns the same number of TXOPs to all nodes in the network.

Medium Access Plan

The medium access plan (MAP-A) contains the scheduling of the transmission slots for a MAC cycle. The duration of the MAC cycle is 40ms (50Hz) or 33,3ms (60Hz) depending on the AC mains frequency. In each MAC cycle, the Spirit HN MAP-A contains four types of slots:

- MAP-A slot: Slot used to transmit the scheduling of the next MAC cycle.
- One or more shared transmission opportunities (STXOP) slots: Slots used to transmit data.
- Inter-domain presence signal (IDPS) slot: Appears as a silence in the MAP scheduling. A signal is transmitted to let neighboring domains determine the MAC cycle alignment as described in the *ITU-T G.9961 Standard (sub clause 8.14.3)*.
- Inter-domain communication channel (IDCC) slot: Slot used to communicate with neighboring domains. It is also described in the *ITU-T G.9961 Standard (sub clause 8.14.3)*.

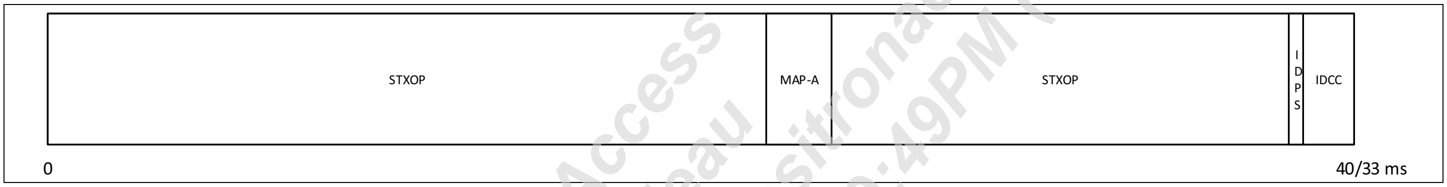


Figure 11: Regular MAC Cycles Scheduling

In certain MAC cycles, other types of slots can be included:

- MAP-D slot: MAP that contains the information required for new nodes to register in the domain or provide information to neighboring domains. The default MAP-D slot appearance ratio is one of each four MAC cycles.
- Registration contention-based transmission opportunity (RCBTXOP) slot: Slot used for new nodes to register in the domain. The default RCBTXOP slot appearance ratio is one of each 11 MAC cycles.
- Inter-domain signaling window (IDSW) slots: Slots used to detect the IDPS from possible neighboring domains to determine the MAC cycle alignment of the neighboring domain as described in the *ITU-T G.9961 Standard (sub clause 8.14.4)*. The IDSW slot appears in three consecutive MAC cycles per second.
- Silence slots: Portions of the MAC cycle where transmission is forbidden for all nodes in the domain. This slot appears only if silence is required in the domain to allow the neighboring domain to transmit or to allow coexistence with other technologies.

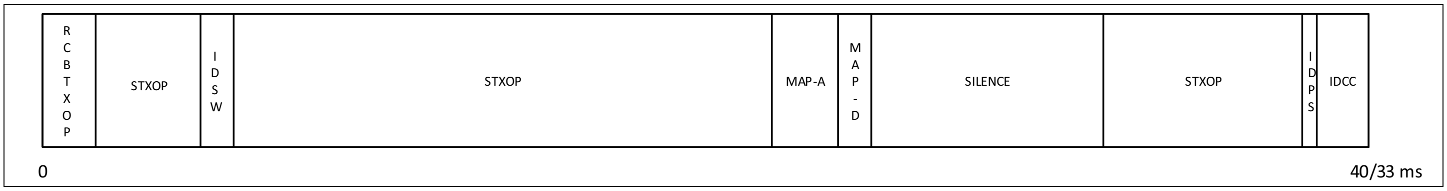


Figure 12: MAC Cycles Scheduling Containing all Possible Slots

Shared Transmission Opportunities (STXOP)

In the Spirit HN software, the STXOPs are assigned to a list of contention-free time slots (CFTS). When a node needs to use its assigned CFTS, it transmits the preamble and the PHY header at the beginning of the CFTS and the other nodes are able to demodulate the PHY header where the duration of the transmission is exposed, so the rest of the nodes know where the next CFTS is placed. When a node has no data to transmit, it lets pass its CFTS without transmitting and the opportunity passes to the next in the list. When the CFTS list is finished, the list pattern is repeated in a circular way.

In the example shown in Figure 13, the MAP specifies that a STXOP must be made of a series of CFTS allocated to devices 1, 2, and 3. Initially, only device 2 has data for transmission and devices 1 and 3 do not use their transmission opportunities. Later, device 1 starts transmitting and device 2 also has remaining data to transmit.

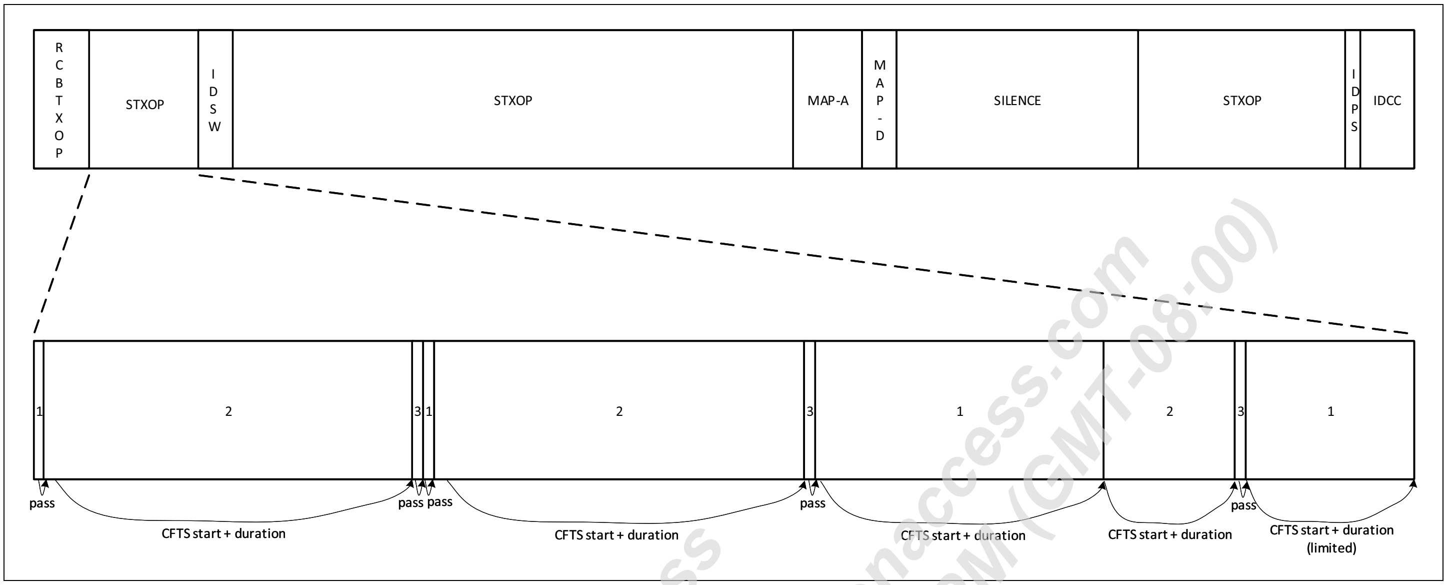


Figure 13: STXOP CFTS Use in Three Nodes Network

In the Spirit HN software, all nodes in the domain have a CFTS in all STXOPs in full visibility domains. When certain nodes have partial visibility of all other nodes in the domain, only nodes with visibility share the same STXOP, as explained in "Partial Visibility Support" on page 42.

Neighboring Domain Interference Mitigation

The Spirit HN software implements Neighboring Domain Interference Mitigation (NDIM) to isolate different G.hn networks. This feature is part of the G.hn *ITU-T Standard* mitigation tool to maximize performance in the neighboring networks multiple dwelling unit (MDU) scenario.

When NDIM is used, every node tracks the domain identifier (DOD) use and automatically selects either the DOD of its own network or an unused one to build a new network. It also resolves the potential collision of different networks using the same DOD.

Domains that use different DODs transmit with orthogonal preambles to ensure that signals in a network are not decoded and are consequently treated as noise. Therefore, the bandwidth is not shared among them unlike other legacy powerline technologies where the effective bandwidth is divided by the number of neighboring networks.

In scenarios where the attenuation between two neighboring domains is very low, the use of orthogonal preambles is not sufficient, because the inter-domain attenuation is comparable to the intra-domain attenuation and therefore, the transmissions of the neighboring domain disturb too much the performance of the local domain. Consequently, the Spirit HN software is able to provide coordination with neighboring domains to maximize overall performance.

The Spirit HN software implements the following mechanisms to achieve NDIM:

- MAC cycle alignment.
- Inter-domain communication.
- Interfering power measurement.
- Neighboring domains coordination.

Note: The Spirit HN software validation tests a limited set of possible scenarios, that is, a maximum of three domains:

- Interfering mode (not coordinated mode).
- Coordinated mode.
- Alignment implicitly and explicitly tested.

In all cases, only one node per domain suffers from high interference.

MAC Cycle Alignment

The NDIM feature must know information about how the domain nodes interfere with the neighboring domains. Information must therefore be exchanged between them.

To share information, neighboring domains must establish a communication channel between them, called inter-domain communication channel (IDCC).

The IDCC must be a contention-based TXOP (CBTXOP), shared by all neighboring domains, and placed at the end of the MAC cycle, so the first step is to align the MAC cycle between all domains.

When all the domains have aligned their MAC cycles and established the IDCC, they all together form a cluster.

Detection of neighboring domains is done using special signals (IDPS) that can be detected setting silent TXOPs to open a special period to detect these signals (IDSW).

The IDPS is always transmitted at the end of the MAC cycle, while the IDSW can be opened at several points called current synchronization points (CSP). These points are located at 0º, 60º, 120º, 180º, 240º or 300º from a zero-crossing point of the AC mains.

The following figure shows the cluster synchronization points in the AC mains cycle.

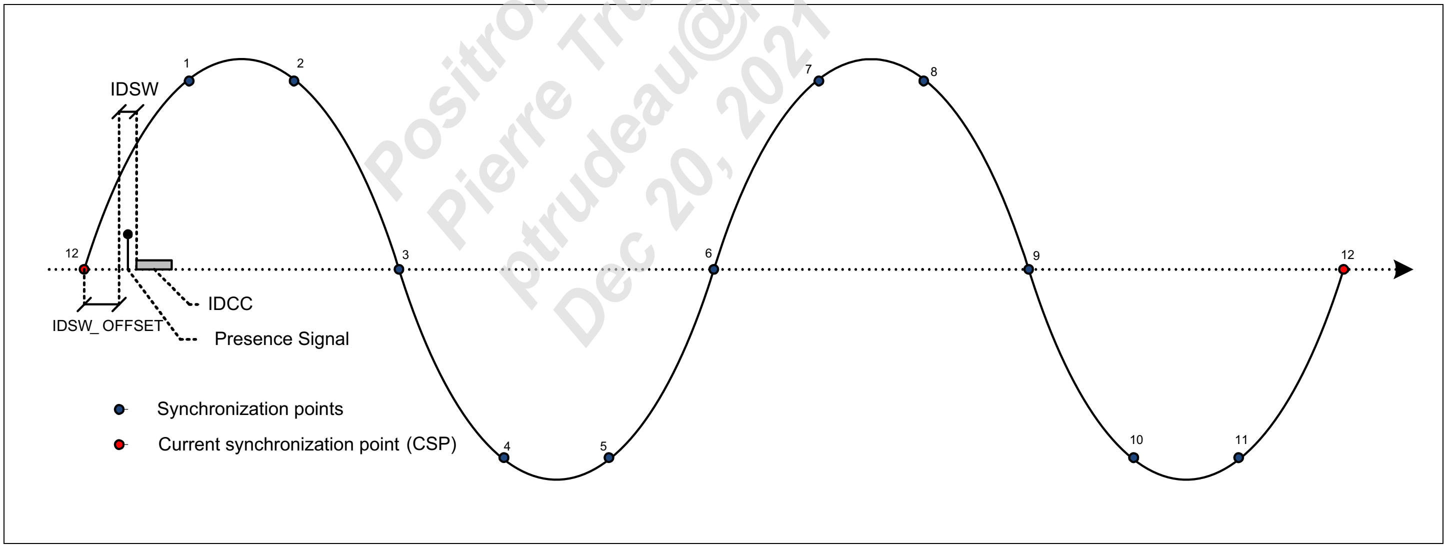


Figure 14: AC Mains Cycle Cluster Synchronization Points

When the IDPS is detected in a IDSW situated at a CSP ! = 0º, which means the detection of a neighboring network with an unaligned MAC cycle, the alignment mechanism must start.

The following figure shows *Cluster 1*, where two domains that see each other have completed the alignment process, and *Cluster 2*, which has not aligned (no visibility with the other cluster or not yet aligned).

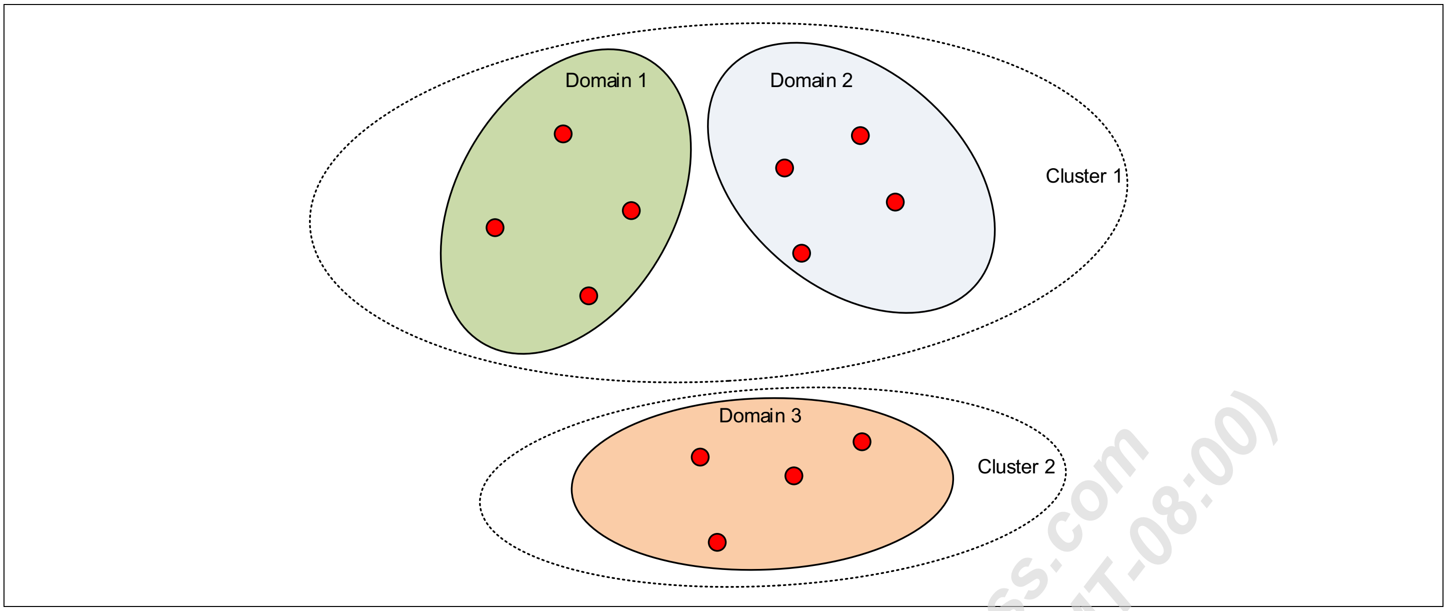


Figure 15: G.hn Domains with Aligned MAC Cycles—G.hn Clusters

The Spirit HN software domain MAC cycle alignment procedure consists of the following steps:

1. Detection of unaligned domain: The neighboring domain IDPS signal is detected in a particular CSP using the periodic opening of IDSW.
2. Precise unalignment measurement: A temporal IDCC called remote IDCC (RIDCC) is opened in the CSP where the IDPS was detected. The MAP-D of both domains are exchanged, in a way that both domains know the synchronization information and clusterID of each domain, so that the MAC cycle alignment offset can be precisely calculated.
3. MAC cycle adjustment decision: The cluster with lower clusterID moves its MAC cycle to align with the neighboring cluster and also inherit a higher clusterID, because both clusters become identical.

Inter-Domain Communication

Once the MAC cycle of the neighboring domains is aligned, all nodes can contend in the IDCC to transmit:

- The MAP-D: It calculates the power received from all nodes.
- Management messages: They coordinate domains with too much interference (high interference).

Inter-Domain Power Measurement

Each node in a domain contends in the IDCC to transmit the MAP-D periodically. These MAP-D are used for the rest of the nodes in the cluster to measure the power received from the node.

Using the received power information, each node can calculate the *worst link* which is the signal-to-interference ratio between the lowest power received from a node in the same domain and the highest power received from a neighboring domain node.

When the worst link is lower than a threshold of 12,5dB, the node considers that it suffers from high interference.

Neighboring Domains Coordination

When a node in a cluster suffers from high interference, the MAC cycle is shared between the domains in the TDMA scheme. The MAC cycle is divided into 16 slots and distributed according to the domain that suffers high interference. Only this domain is able to transmit in the slots that are assigned to it. The number of slots assigned for each domain depends on the number of interfering domains in the cluster, assigning for each domain 1/2, 1/3, 1/4, etc. of the slots.

The NDIM coordination protocol exchanges the required messages using the IDCC between the neighboring domains to distribute the MAC cycle.

The following figure shows the 16 MAC cycle slots distributed to three coordinated domains.

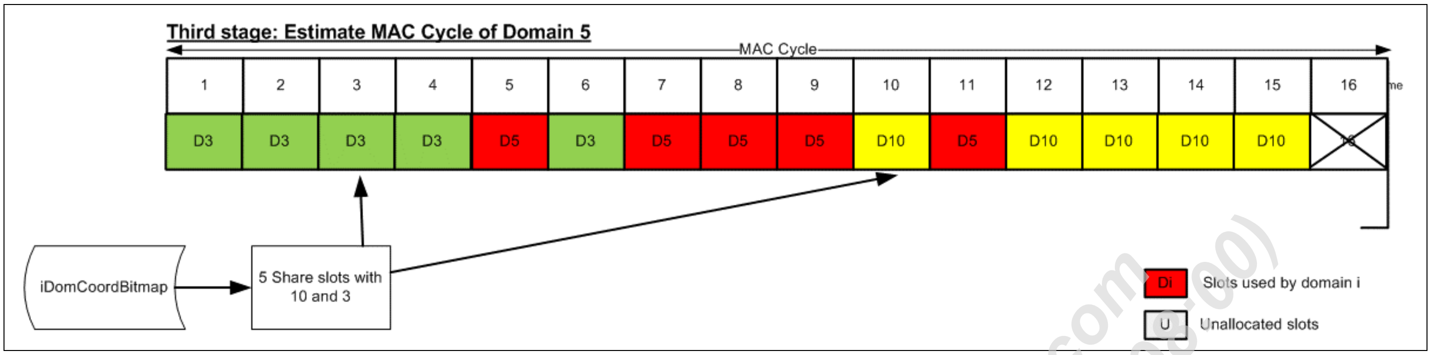

Third stage: Estimate MAC Cycle of Domain 5
MAC Cycle
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
D3 D3 D3 D3 D5 D3 D5 D5 D5 D10 D5 D10 D10 D10 D10
iDomCoordBitmap
5 Share slots with 10 and 3
Di Slots used by domain i
U Unallocated slots

Figure 16: G.hn MAC Cycle Slots Assigned for Three Coordinated Domains

The following table lists the main NDIM configuration parameters. Certain are used to configure the mode of operation, while others provide information on the current detection and coordination with them.

Table 29: NDIM Configuration and Information Parameters

New Configuration Layer Parameters	Description
`NDIM.GENERAL.COMMON_CHANNEL_ENABLED`	Enables or disables the IDCC scheduling.
`DIM.GENERAL.CLUSTERID`	ClusterID to which the node belongs. Useful for checking that domains are aligned.
`NDIM.GENERAL.N_VISIBLE_DOMAINS`	Number of visible neighboring domains.
`NDIM.GENERAL.COORDINATION_APPLIED`	The DM is coordinated with the neighboring domain.
`NDIM.INFO.NEIGHB_NETWORK_DODS`	List of the DoDs used by the detected neighboring domains.
`NDIM.INFO.PEER_NODES_POWER`	Lists that contain the tuples of the same domain nodes deviceID and received power.
`NDIM.INFO.NEIGHB_NODES_POWER`	Lists that contain the tuples of interfering domain DoD, deviceID and received power.
`DIM.INFO.NEIGHB_DOMAIN_NAME`	List of the neighboring domain names.
`NDIM.INFO.NEIGHB_COORD_DOMAINS`	List with the DoD of the coordinated neighboring domains.
`NDIM.INFO.NEIGHB_HIGH_INTERF_NODES`	List that contains the tuples of the DoD and deviceID causing high interference.
`NDIM.INFO.NEIGHB_COORD_ALLOCATION`	List with the DoD assigned to each of the 16 MAC cycle slots defined for coordination.

Coexistence with Legacy PLC

This component must ensure good coexistence with legacy PLC devices (non-compliant with G.hn) so that the MaxLinear PLC devices provide a spare-time channel for legacy network activity within the powerline channel.

Interference caused by neighboring networks is not strong enough to produce noticeable effects in performance. The main issue is when legacy and the MaxLinear devices are plugged in the same house, so interference is high.

The MaxLinear devices detect the presence of legacy PLC devices that send hello frames via Ethernet. In that case, the prerequisite is that all devices are connected to the same LAN. When a legacy device is detected, all G.hn networks enter in coexistence mode.

Table 30: Parameters Related to Coex Functionality

Parameter	Description

`COEX.GENERAL.ENABLE_DETECTION`	Set to *YES* to enable detection of alien networks. Set to *NO* to disable it. Reset after change.
	The recommended configuration in the field for automatic coexistence detection is:
	`COEX.GENERAL.ENABLE_DETECTION = YES`
	`COEX.GENERAL.ENABLE_MODE = NO`
	`COEX.GENERAL.ENABLE_MODE_DEPENDENCY = YES`

`COEX.GENERAL.ALIEN_NETWORK_TIME_PERC`	GPIO connected to the external watchdog (if present)
`COEX.GENERAL.ALIEN_NETWORK_TIME_MAX`	Maximum percentage of time respected for the alien network (in quantum of 2%). The maximum allowed value is 70. When MAX > MIN, the time assigned to the alien depends on the traffic needs of the G.hn network. This is done automatically by the firmware. In this case, set this value to *50*.
`COEX.GENERAL.ALIEN_NETWORK_TIME_MIN`	Minimum percentage of time respected for the alien network (in quantum of 2%). MaxLinear does not recommend values lower than 20. For example, if set to *20*, at least 20% of the time is respected for the alien network, regardless of the G.hn traffic needs. This percentage increased if G.hn does not have heavy traffic.
`COEX.GENERAL.ALIEN_NETWORK_TIME_FIX`	Fixed percentage of time for the alien network. The value 0 means that the time of the alien network is automatically assigned. MaxLinear does not recommend values lower than 24. If this parameter is set, `COEX.GENERAL.ALIEN_NETWORK_TIME_MAX` and `COEX.GENERAL.ALIEN_NETWORK_TIME_MIN` are bypassed. This parameter is intended to allow an external entity to control the percentage of time assigned to G.hn.

Quality of Service (QoS)

The Spirit HN software implements:

- Strict priority policies for packet prioritization in transmission queues.
- QoS policies that are used to decide the channel time allocation.

The Spirit HN software allows the configuration of packet classification rules for prioritization. The default configuration includes classification by DSCP and optionally by *IEEE 802.1p* VLAN priority.

IPv4 DSCP prioritization is applied before VLAN prioritization when the IPv4 protocol is detected. To classify according to the VLAN rule, you must disable the IPv4 rules.

You can use the configuration layer parameters to configure the settings related to this feature, such as enable/disable DSCP classification or VLAN classification. You can use the `PACKETCLASSIFIER` group of the configuration layer parameters for this purpose.

Statistics on the traffic received by a node are available. Information on the number of frames and FEC blocks received per node, and the percentage of errors since the last reset of the statistics are collected. The statistics can be reset to start a monitoring period.

For more information about the link statistics, refer to the `FLOWMONITOR` parameters described in the configuration layer parameters html document delivered along with the firmware.

Buffer Management Policies

In the 88LX515x, two types of packet buffers can be assigned to each connection:

- CBR buffers: Committed buffers. The memory is exclusively assigned to a connection. This memory is not used by other connections even if other connections have exhausted their buffers.
- VBR buffers: Shared buffers. The memory is assigned to a connection that can be shared with other connections that are allowed to use VBR buffers. If these types of buffers are assigned, it does not necessarily mean that they will be used. It depends on whether they are available when a connection needs to use them.

The dynamic buffer management uses two different policies to manage the distribution of the existing buffers, depending on the type of buffer:

- CBR buffers: Defined by a policy that divides the CBR buffer pool between the number of established data connections. The `QOS.QUEUEMANAGEMENT.CBR_POLICY` parameter controls the CBR buffer pool distribution. The value allowed for the Spirit HN is *EVENLY*.
- VBR buffers: Defined by a policy that assigns a percentage of the total VBR buffer pool to each established data connection. The `QOS.QUEUEMANAGEMENT.SATURATION_POLICY` parameter controls the total VBR buffer pool distribution. The value allowed for the Spirit HN is *MAX_MEMORY_USE*.
  The `QOS.QUEUEMANAGEMENT.MAX_MEMORY_USE` parameter determines the percentage amount of the total VBR pool assigned to each data connection.

The following table lists the corresponding configuration parameters.

Table 31: Buffer Management Configuration Parameters

Parameter	Description
`QOS.QUEUEMANAGEMENT.CBR_POLICY`	Selects the CBR policy. The policy determines how to share the CBR buffer pool (committed buffers) among the data connections. The available values are: ■ EVENLY (default). ■ BWLIMIT (not supported in home networking).
`QOS.QUEUEMANAGEMENT.MAX_MEMORY_USE`	Maximum percentage amount of the total buffer memory for packets in one data connection. The default value is 100%
`QOS.QUEUEMANAGEMENT.SATURATION_POLICY`	Selects the SATURATION policy. The policy determines how to share the VBR buffer pool (shared packet buffer) among the data connections. The available values are: ■ MAX_MEMORY_USE (default). ■ BWLIMIT (not supported in home networking).

Caution: The default configurations for the buffer management parameters have been determined to maximize network performance. Do not modify them without MaxLinear's supervision.

MAC Scheduling Power Saving (MSPS)

The G.hn *ITU-T G 9961 Standard* defines the following basic types of power saving.

- Normal mode (L0): This is the mode in which transmission up to the maximum data rate is possible without any power saving.
- Efficient-power mode (L1): In this mode, power consumption is reduced by limiting medium access for transmission and reception only to a portion of a MAC cycle. This relies on the short inactivity schedule mechanism and the inactivity time can be different in each node.
- Low-power mode (L2): In this mode, power consumption is reduced by suppressing medium access over multiple MAC cycles up to 36000 MAC cycles. Only a limited data rate is supported. The procedure that controls this is the long inactivity schedule. Different nodes can request different periods for power saving, but all must have a common active time.
- Low-power mode (L3): In this mode, power consumption is reduced by suppressing medium access over multiple undetermined MAC cycles. Only a limited data rate is supported (see the *sub clause 8.3.6.1.1 Long inactivity scheduling* of the *ITU-T G.9961 Standard)*. In this mode, the node can be woken up by any node of the domain or by new transmissions through the A interface.
- Idle mode (L4): In this mode, power consumption is minimized by suppressing any activity related to the domain. The node is switched on and connected physically to the home network, but no data or control message is transmitted or received. A node can enter this mode when no data can be transmitted/received through the A interface (Eth, SDIO). Remote wake-up is not supported.

Transitions between these modes are always done through the L0 mode and they all involve a sort of negotiation between the nodes and the domain master (DM). Typically, a node requests to move to one of the mentioned power saving modes in the DM. The DM assesses the impact of the request for the domain and accepts or rejects it. The scheduling is modified in the MAP according to the decision, and the nodes also apply any required changes. All nodes in the domain know the power saving state of the rest of the devices.

Currently, the Spirit HN software only supports the following modes:

- L0 mode: It is a full performance mode without any power saving.
- L4 mode: For more information, see "Standby Modes" on page 88.
- L1 power saving mode: It is included in the Spirit HN as MAC scheduling power saving.

When the L1 power saving feature is enabled, the modem is allowed to be in idle mode, thus saving power during certain periods of the MAC cycle.

For this purpose, the MAC cycle is divided into 1/16th slots and each end point (EP) can request to the DM—or the DM itself—which slots to use for power saving. This decision is made based on traffic, temperature, or any other criteria that can be taken into account. The EPs make the request by sending an `IAS_ShortInactivity.req` message.

The DM assesses whether the request can be granted and tries to adapt the slots to the current scheduling and previous requests from other nodes in the domain.

The selected slots (*short inactivity schedule* as named by the ITU) for the L1 power saving are only published in the MAP-A messages, using the short inactivity schedule MAP auxiliary field.

You can enable this feature for any media (PLC, phone, or coax).

The following figure shows the basic L1 power saving message sequence.

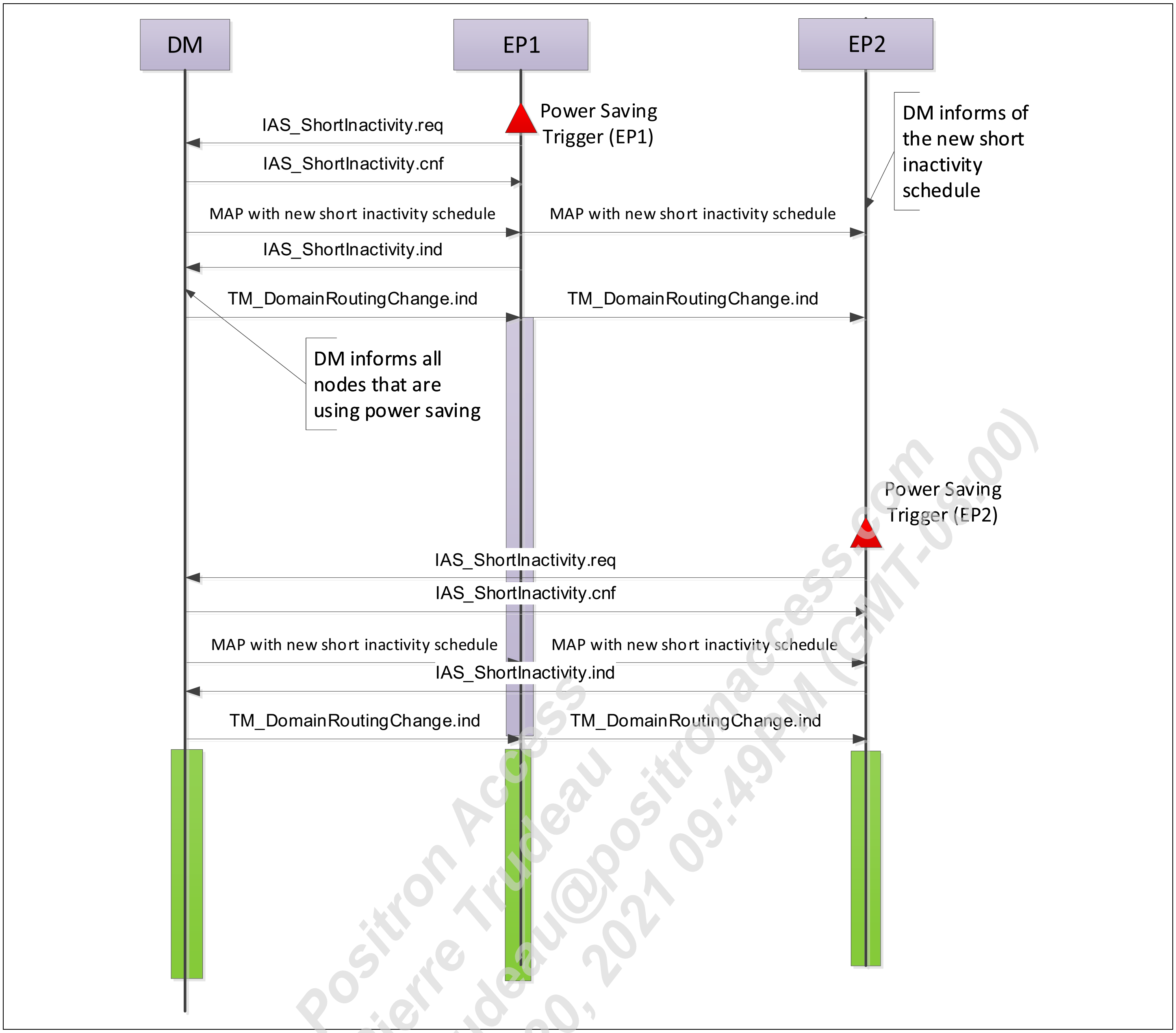

DM
EP1
EP2
Power Saving Trigger (EP1)
DM informs of the new short inactivity schedule
IAS_ShortInactivity.req
IAS_ShortInactivity.cnf
MAP with new short inactivity schedule
MAP with new short inactivity schedule
IAS_ShortInactivity.ind
TM_DomainRoutingChange.ind
TM_DomainRoutingChange.ind
DM informs all nodes that are using power saving
Power Saving Trigger (EP2)
IAS_ShortInactivity.req
IAS_ShortInactivity.cnf
MAP with new short inactivity schedule
MAP with new short inactivity schedule
IAS_ShortInactivity.ind
TM_DomainRoutingChange.ind
TM_DomainRoutingChange.ind

Figure 17: Basic L1 Power Saving Message Sequence

MSPS Operation

Triggers

The amount of power saving is configured according to different triggers. There are mainly three types of triggers:

- Triggers based on temperature (to protect the device). They are disabled by default.
- Triggers based on priorities.
- Triggers based on traffic needs.

The different triggers are, in order of priority:

- Reset triggers: If enabled, when the temperature is higher than the configured value for more than the configured time, the device is reset to prevent damage. The L2 configuration parameters are in the `MSPS.RESET` subgroup.
- Forced triggers: Only for validation. The `MSPS.L1.ACTIVE_SLOTS_FORCED` L2 configuration parameter forces a request with the configured number of active slots. Any other trigger is ignored. It is intended for validation purposes only.
- Temperature-based externally-triggered throttling: Controlled externally, this mechanism allows the third-party SoC to reduce the maximum percentage of time in active mode in order to reduce the temperature of other chips in the same system. The maximum active slots value is `MSPS.EXTERNAL.ACTIVE_SLOTS_MAX` but the device can request less depending on the other triggers.
- Temperature-based internally-triggered throttling: This mechanism examines an internal 88LX515x temperature sensor and reduces the maximum percentage of time in active mode in order to reduce the internal temperature.
  This mechanism ignores traffic and focuses only on ensuring that the chip is not damaged by high temperature.
  The `TEMPSENSORS.GENERAL.MEASURE` L2 parameter can check the sensor temperature. It returns a value in (1/100)°C units with an accuracy of ±4°C.
  As the temperature changes slowly, it is checked every `MSPS_TEMPERATURE_CHECK_PERIOD_ITERATIONS` of the trigger period.
- Priority detection: If the traffic of one or more of the configured priorities is detected, immediately exit from power saving to ensure low latencies. Therefore, power consumption increases.
  Temperature-based throttling is still applied to prevent damages (if enabled).
  The MSPS L2 configuration parameters are in the `MSPS.PRIORITY` subgroup.
  To configure the priority detection itself, there are three methods:
  - DSCP support. For more information, see “DSCP Support” on page 64.
  - VLAN. For more information, see “VLAN” on page 60.
  - Custom rules. For more information, see “Custom Rules” on page 64.
- Minimal state: This trigger configures a minimal value of active slots when there is a very low traffic detected in the Ethernets, only to maintain the network.
  It is the state with maximum power saving.
  The traffic is checked with the triggers period and it is configurable through the `MSPS.MINIMAL` L2 parameters.
- Traffic-based power saving: Based on the actual traffic, the firmware tries to minimize the percentage of time in active mode with minimal impact in performance. Latency is always increased when reducing the number of active slots.
  Temperature-based throttling is still applied to prevent damages (if enabled).
  The L2 configuration parameters are in the `MSPS.THROUGHPUT` subgroup.
  With this configuration, four basic scenarios related to traffic-based power saving are supported:
  - Minimal state: State when there is very low traffic in the Ethernet, with maximum power saving.
  - Active cycle 50%: This is the minimum value of the active slots when non-minimal traffic is detected. This minimum value is set to *50%* to avoid high latencies.
  - Active cycle 50%–100%: Depending on the traffic, the number of active slots adapts. This is not a lineal adaptation and it is better to avoid packet loss or reduce performance.
  - Active cycle 100%: When all nodes in the network work at 100%, performance is better than in any other case. This is because the MAC efficiency is reduced when using the MSPS because a higher number of STXOP is required.

Remarks

- When the number of slots requested by all nodes in the network is 16, the MAC cycle is divided into two or three STXOPs, with maximum performance.
  When the number of slots requested by at least one node in the network is less than 16, the MAC cycle is divided into eight STXOPs, and one or more of them are used to enter one or more nodes in power saving. Each STXOP has a size equivalent to two active slots. For this reason, `MSPS_ACTIVE_SLOTS_STEP` is set to *2*. In this case, performance is affected because increasing the number of STXOPs in a MAC cycle reduces efficiency.
- When the number of slots requested by at least one node in the network is less than 16 and there are more than eight nodes in the network, the MAC cycle is divided into four STXOPs due to current MAC limitations which cannot properly manage such large MAPs. In this particular case, only the minimal state, 50% and 100% of activity, is supported.
- Any change in the traffic requires the maximum possible value of active slots, to avoid packet loss or impact in performance. This includes start/stop traffics.
- Bursty traffic, such as the traffic when there is a video on demand, can be detected. If this feature is not included, with bursty traffic the number of slots is always the maximum due to changes in the traffic.
- If legacy nodes are present in the network, the DM disables the MSPS functionality because the legacy nodes are not ready to avoid transmissions to idle nodes.

Table 32: MSPS Configuration Parameters

Parameter	Description
`MSPS.EXTERNAL.ENABLE`	Enables or disables the external trigger feature.
`MSPS.INTERNAL.ENABLE`	Enables or disables the internal trigger feature.
`MSPS.L1.ACTIVE_SLOTS`	Current active slots in a MAC cycle.
`MSPS.L1.ENABLE`	Enables or disables the support for L1. Versioning reports this value as a G.hn ITU low power capability.
`MSPS.MINIMAL.ACTIVE_SLOTS`	Active slots to configure when the node is in minimal state.
`MSPS.MINIMAL.ENABLE`	This parameter enables or disables the detection of reduced Ethernet traffic to set a minimum active slots configuration. Latency in this minimal mode is not guaranteed.
`MSPS.RESET.ENABLE`	Enables or disables the reset due to the high temperature feature. It is intended to prevent damage due to high temperatures.
`MSPS.RESET.MS`	When the node has a temperature higher than `MSPS.RESET.TEMP` during `MSPS.RESET.MS`, it is reset to prevent damage.
`MSPS.RESET.TEMP`	When the node has a temperature higher than `MSPS.RESET.TEMP` during `MSPS.RESET.MS`, it is reset to prevent damage.
`MSPS.THROUGHPUT.ENABLE`	Enables or disables the traffic-based throughput trigger feature.

Partial Visibility Support

This feature has been validated only for the HN powerline (not coax nor phone).

Partial Visibility Issue Description

Currently, in HN scenarios, all nodes present in a domain network establish connections with each other (mesh topology). In addition, the domain master (DM) includes all nodes of the domain network in each STXOP. As a result, it is necessary that all nodes have sufficient link quality between them to properly receive PHY headers, as shown in the following figures.

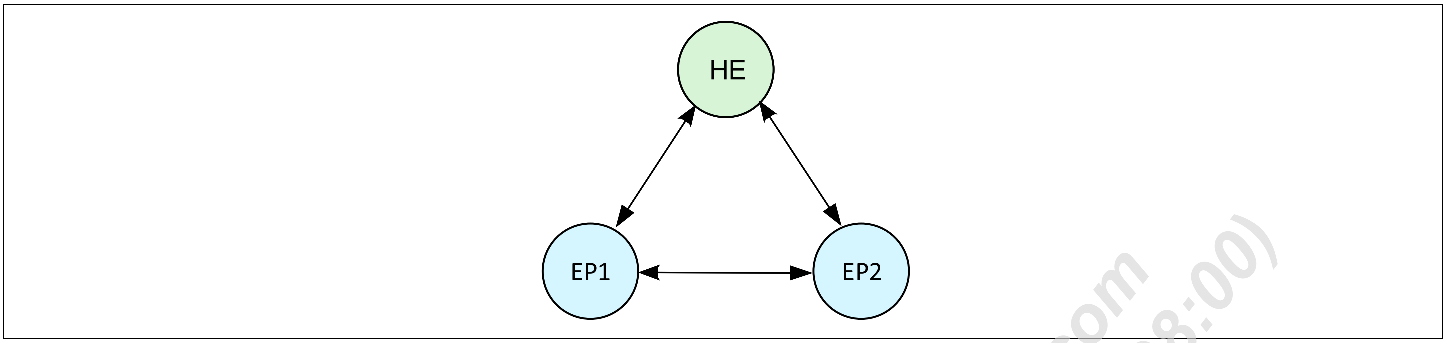
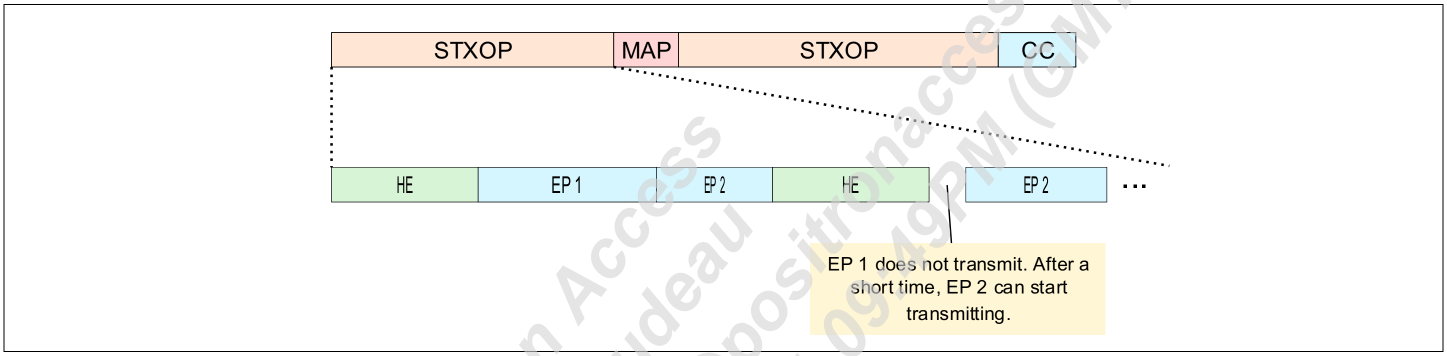


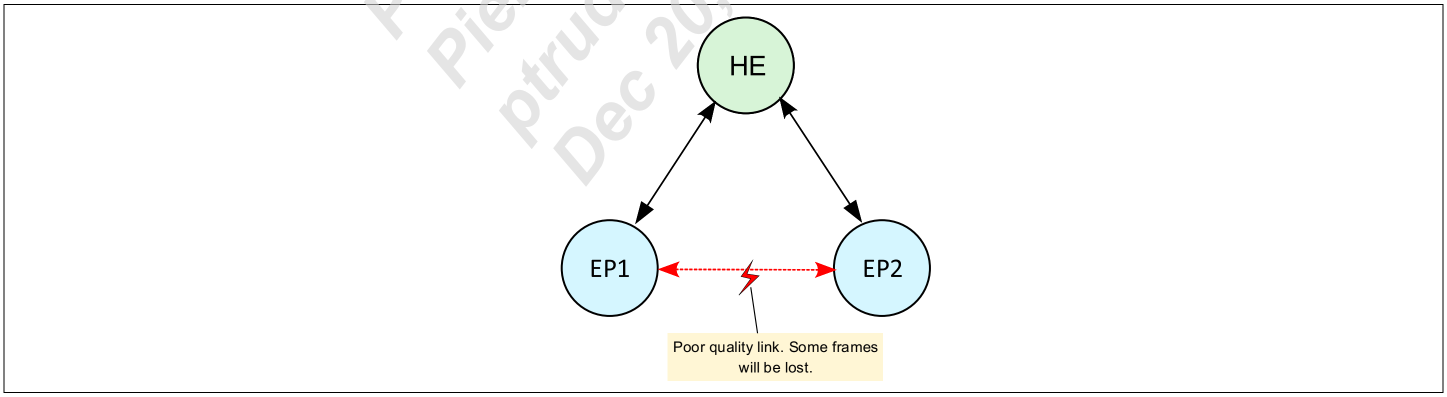
Figure 18: Home Networking Mesh Network



EP 1 does not transmit. After a short time, EP 2 can start transmitting.

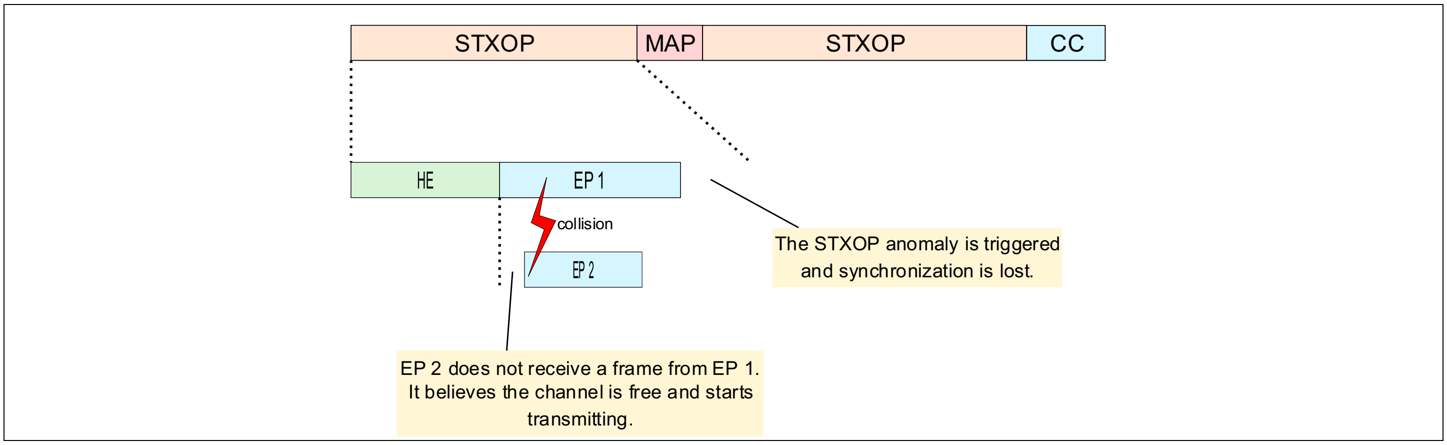
Figure 19: STXOP with Good Links

However, in certain scenarios, with high attenuation or noise, this requirement cannot be met. In this case, the transmissions within the STXOP collide between different nodes, as shown in the following figures.



Poor quality link. Some frames will be lost.

Figure 20: Home Networking Network with Poor Link



The STXOP anomaly is triggered and synchronization is lost.

EP 2 does not receive a frame from EP 1. It believes the channel is free and starts transmitting.

Figure 21: STXOP with Poor Link

Communication is very limited in this type of scenario when partial visibilities are not supported.

Partial Visibility Objectives

Prerequisite

- The nodes must be able to, at least, register with the domain master.

Partial Visibility Detection

- The node must estimate the quality of the link with each possible destination to decide if a direct link can be established.
- The link quality estimation is performed even if the remote node does not transmit to the local node.
- The link quality estimation is an ongoing process as it can change over time.
- The link quality estimation is based on PSD and NOISE.
- The link quality estimation must disrupt the network as little as possible.

MAC Scheduling

- The DM must take into account the visibilities in the domain network when building the MAC scheduling, in order to avoid collisions.
- This feature must take into account the MSPS, NDIM, and COEX specific MAC scheduling.
- The DM must adjust the transmission opportunities of each node according to the traffic requirements.

Routing

- Peer-to-peer traffic (without a direct link) must be routed through the DM (using it as a data relay).

MAC Scheduling

In previous firmware versions, all nodes in the domain were included in all STXOPs of each MAP cycle. To support partial visibilities, it is necessary to add management to decide which nodes must be included in each STXOP to avoid BLER and STXOP anomalies.

The following figure shows the inputs and outputs of the algorithm.

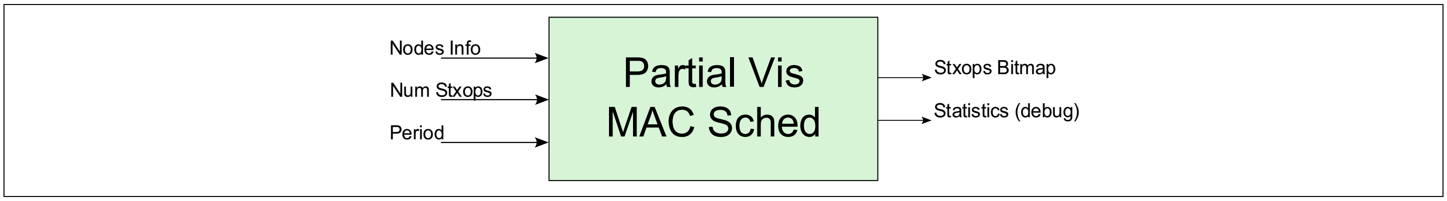


Figure 22: MAC Scheduling—Inputs and Outputs

- Nodes Info: It includes the visibility list of each node, its requested number of STXOPs to meet its traffic requirements, and the maximum number of STXOPs per cycle in ACTIVE state (from the MSPS).
- Num Stxops: Number of STXOPs per MAP cycle. Typically, it is set to *8*.
- Period: The algorithm tries to meet the traffic requirements of each node during this period, distributing the nodes on each STXOP as required.
- Stxops Bitmap: Information required to locate the nodes to insert in each STXOP.
- Statistics: Statistics for debug purposes.

Cycle Partitioning

Cycle partitioning depends on several elements such as the MSPS, COEX, NDIM, etc. Typically (when nodes are active), the cycle has two big STXOPs. However, when **partial visibilities are enabled and indirect nodes are detected**, the cycle must be split as shown in the following figure.

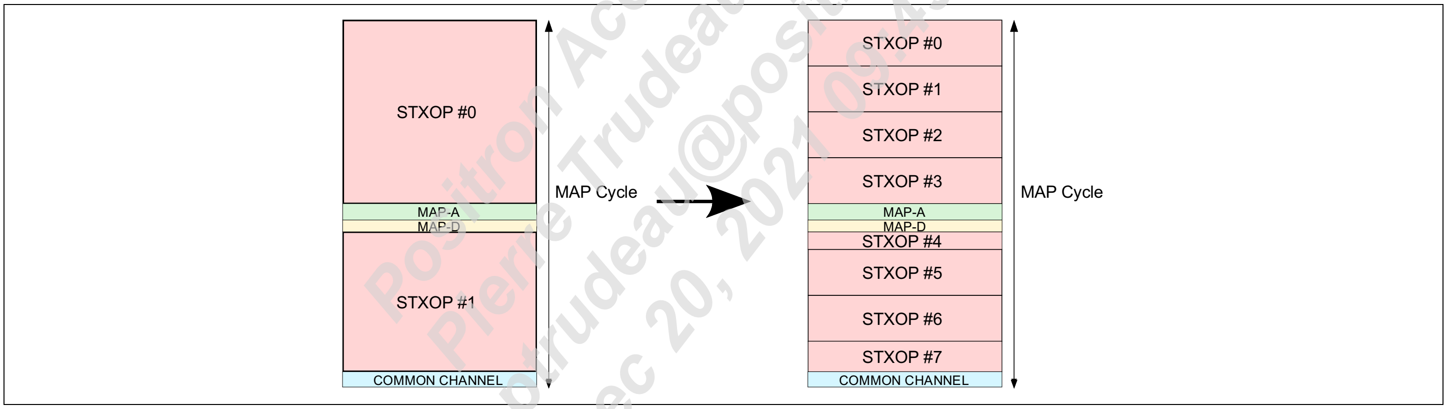


Figure 23: MAC Scheduling—Cycle Partitioning

This is the same partitioning as the one used for the MSPS feature, when certain nodes become idle. This split can be modified by COEX or coordinated NDIM, as these features include a big SILENCE in the cycle, removing certain of those STXOPS. For the partial visibilities algorithm to work, it is necessary to know:

- The number of STXOPs in the cycle.
- The requested number of STXOPs for each node.

The DM analyzes the frames sent in the last MAP cycles and assigns the time slot (TS) according to the activity detected for each node, avoiding placing two nodes with lack of visibility in the same STXOP, thus avoiding collisions.

Routing

When end point (EP) nodes in a domain do not have direct visibility, it is necessary to reroute all traffic between them through the DM. All EP nodes have direct visibility with the DM.

For this purpose, the RTM block and the bridge forwarding table (BFT) have been modified to allow the relay of data between EPs through the DM.

Table 33: Partial Visibility Configuration Layer Parameters

New Configuration Layer Parameters	Description
`TOPOLOGYMANAGER.VISIBILITY.CHECK_ENABLE`	Enables or disables the visibility check for hidden nodes.
`TOPOLOGYMANAGER.VISIBILITY.INFO`	Last input information provided to the visibility schedule algorithm. This information is only valid if the node acts as a DM. If the node is an EP, this information is not valid.
`TOPOLOGYMANAGER.VISIBILITY.PARTIAL_DETECTED`	TRUE if partial visibilities are detected between the nodes in the domain. This information is only valid if the node acts as a DM. If the node is an EP, this information is not valid.
`TOPOLOGYMANAGER.VISIBILITY.NUM_CARR_THR`	Number of carriers whose signal-to-noise ratio (SNR) is greater than the SNR threshold to consider that a node has visibility.
`TOPOLOGYMANAGER.VISIBILITY.SOFT_DECISION_WEIGHT`	Weight to apply in the soft decision. It indicates how many times the samples under the threshold are more weighted than those above.

G.hn Data Link Layer

This section describes the data link layer (DLL) features implemented in the Spirit HN software.

Encryption

The Spirit HN supports encryption of all data connections using a single encryption key. It uses the AES-128 encryption algorithm, as specified in the *ITU-T G.9961 Standard (sub clause 9.1)*, allowing the use of a variable-size message integrity code (MIC).

In the current implementation, all nodes in the domain use the same encryption key. Encryption can be enabled in several ways:

- By performing a pairing procedure to enter/generate a secure domain. In this case, the node that have the role of secure domain master generates a random domain name and encryption key and provides them to the accepted end point (EP) nodes.
- By setting a password in all nodes that need to communicate, through the `PAIRING.GENERAL.PASSWORD` configuration layer parameter.

Passwords are not stored in plain text, but always as a hash in flash.

MaxLinear recommends the *IEEE 802.1X* authentication protocol. For more information about its implementation, refer to the *G.hn Spirit Software External Authentication Application Note* (103AN).

Domain Master Selection Protocol

The Spirit HN software implements the Domain Master Selection Protocol specified in the *ITU-T G.9961 Standard (sub clause 8.6.6)*.

The selection of the domain master (DM) is automatic because all nodes have the ability to be domain master.

The domain master backup functionality is not yet supported. When a node is registered in a domain and the DM fails, each node in the network must follow the procedure described in the *sub clause 8.6.6.2* of the *ITU-T G.9961 Standard* to recover the network. The procedure describes how an end point previously registered with a DM that is no longer available becomes the new DM and the rest of nodes from the previous domain register with this new DM.

The *sub clauses 8.6.6.3 Ranking of domain masters capabilities and 8.6.6.4 Handing domain master's role to a more capable node* of the *ITU-T G.9961 Standard* are not yet supported by the Spirit HN software. However, as soon as a DM detects another DM transmitting in the same domain, it closes the domain and establishes a new one.

The master selection component drives the search for an existing DM and triggers the decision to become a DM of a new domain in the event that a suitable candidate is not found.

Although there is a component named *master selection*, the process is complex and many components have a role in the procedure. The main components involved are:

- Master selection: The main component that deals with the sequence to follow during the process. It manages the T0, T1, and T2 different periods and manages the reception of the MAPs to obtain the relevant information and take the necessary measures. For more information about T0, T1, and T2, see "Become a Domain Master (DM)" on page 49.
- Topology manager: It manages the collection of information from all DM candidates and in general the selection of the best one to use as a DM.
- Sync: It receives timing information from the MAPs, to calculate the adjustments required for frequency offset and tracking.
- Network Admission Protocol (NAP): It is used for all the processes related to the registration of modems.
- Pairing: It manages the entire sequence related to the selection of a secure DM and the admission of new nodes in a secure domain.
- Preamble: It deals with the setting of the correct seeds to use in the Rx and Tx.

Search for a Suitable Domain Master (DM)

Figure 24 on page 48 shows the process followed by a node that just booted, to search for a new DM. There are several domains, each with its own DM.

The figure shows different elements that are involved in the process and the states they go through until a successful registration takes place.

An EP searches for a DM that uses the same domain name that it has assigned or, in the case of pairing, a secure DM that accepts the node and provides a valid domain name and encryption key during the process. If after a time T0 + T1—defined by the *ITU-T G.9961 Standard* and where T0 is 10 seconds and T1 is a random time from 0 to 1 second—, no DM meets these requirements, the node becomes a DM.

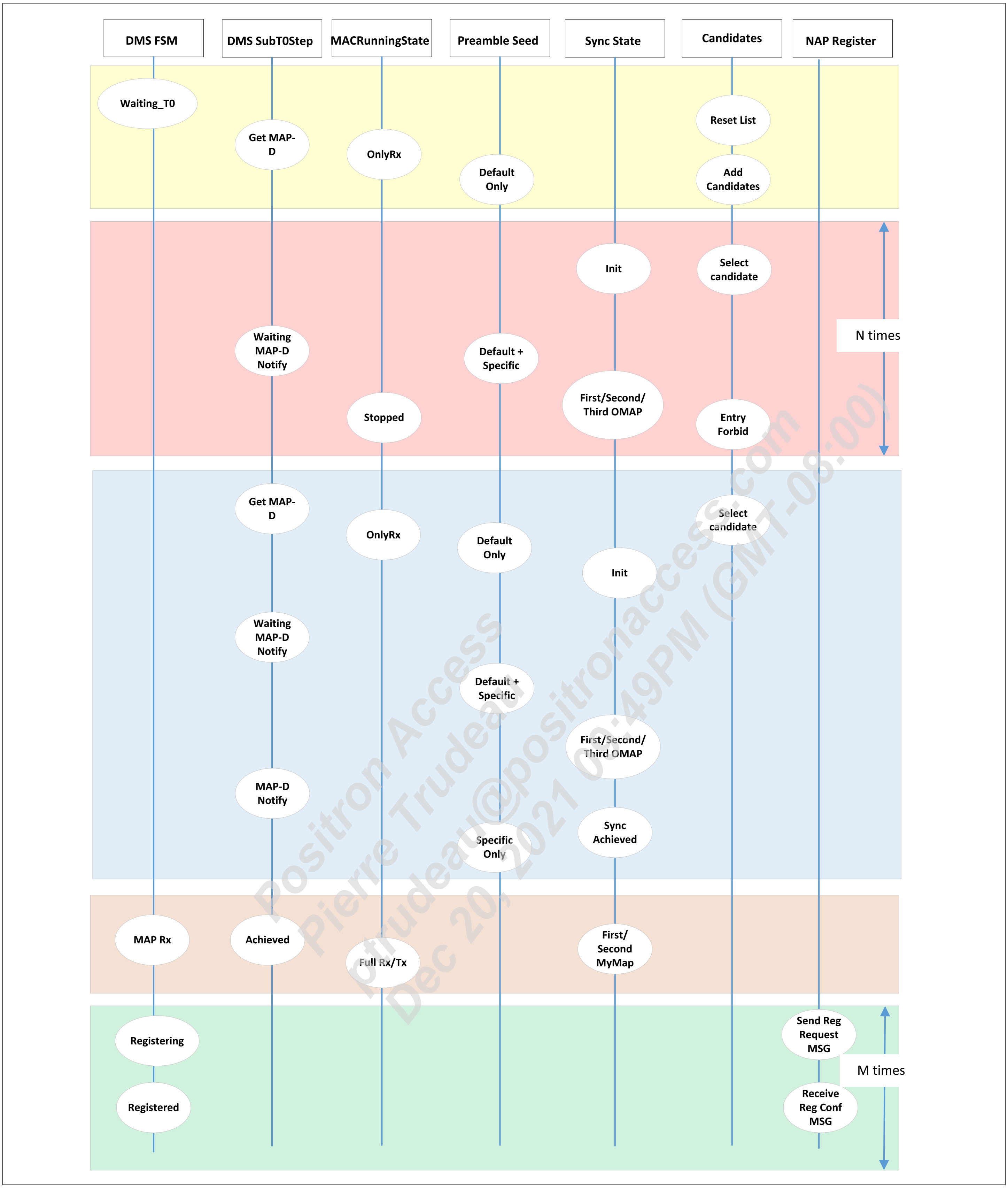


Figure 24: **Domain Master Search Process**

In Figure 24 on page 48, there are five phases:

- Phase 1 (yellow): The EP node only listens for candidates and adds them to the list of candidates. It also measures the channel frequency response (CFR) and power for them in order to select the best the first time. The default seed (the one that the MAP-D uses and that any node can listen to) is used to start receiving MAP-Ds from the different DMs.
- Phase 2 (pink): A candidate node is selected and the synchronization phase takes place. In this example, the selected node is not valid (it does not match the DM or it is not accepted during a pairing operation) and is therefore marked as forbidden. Consequently, the process is restarted with a different candidate. This process can occur 0 times (if the first node selected is the right one) or as many times as there are candidates (if none of them are suitable). In the figure, *N* represents the number of times this phase takes place. The initial synchronization is performed using a hardware block called FERC, which can rotate the QAM constellation to compensate the synchronization offset within certain limits. A synchronization firmware component performs the necessary timing reference calculations and the continuous tracking of the synchronization, based on the timing information obtained from the MAP-A PHY header reception.
- Phase 3 (blue): This is the same phase as phase 2, but it ends with a successful synchronization with the suitable DM. During this phase, the node stops listening to the default seed and starts listening to only the specific seed (except during the common channel, where any domain can transmit).
- Phase 4 (brown): Once the initial synchronization is complete and the MAP-D payload is decoded, the synchronization continues to iterate for better accuracy. Poor synchronization decreases communication performance, limiting the maximum frame length that a node can decode without error and the bit load that can be used. Once this is achieved, the MAP-As are fully decoded and the MAC is allowed to start transmitting.
- Phase 5 (green): Once the MAC is allowed to transmit, the registration request message can be sent. The more registration slots available, the faster this process. It is important that the DM processes the first registration request correctly to prevent retries, which adds long timeouts to the registration process. These retries can take place, for example, if the node of the DM is still registered when the DM receives the registration request, which can happen if the EP leaves the domain and reenters very quickly. In the figure, *M* represents the number of retries made before a successful registration.

Several of these phases have been optimized to reduce the time to a minimum.

Become a Domain Master (DM)

This section describes what happens when a node cannot find a suitable candidate to become a DM.

As described in the *ITU-T G.9961 Standard*, when a node starts to search for a suitable DM, it initiates a timer T0, with 10 seconds as the default value. During this time, the process continues until T0 expires or a suitable DM is found. While T0 still runs, the node continues to try and add new candidates.

Once T0 expires, T1 starts, which corresponds to a random time between 0 and 1 second. It is only after this time elapses that the node becomes a DM, selecting a free domain identifier (DOD) and starting to transmit MAP-As with the specific seed associated with the selected DOD. T1 is a random time to minimize the probability that several nodes become DM at the same time if many are powered on at the same time (after a power blackout, for example).

The MAPs sent must be aligned with the MAPs sent from other possible DMs. This means that all of them must start at the same time (with a certain tolerance) for the common channel to fall in the same place for all MAPs from different domains.

After T1 expires, a third period T2 starts. This period lasts approximately 400ms. While T2 runs, a DM continues to send MAPs, but does not accept any EP in the domain. It monitors for new DMs that were not in the candidate list but suddenly appear in the medium and in case any are found, it stops being a DM and starts the whole process again. The purpose is to prevent the formation of multiple independent domains due to several nodes becoming DM at the same time. This is particularly important in the case of pairing processes where a new secure domain is created, because new secure domain masters generate their domain names during the pairing operation.

After T2 expires, the DM allows the admission of new nodes into the domain and the process to become a DM ends.

A DM always monitors the domain to detect the presence of other DMs using the same DOD, in which case it drops as a DM. In that case, another (or the same) node becomes the new DM (in a different DOD) and the domain is reestablished.

Network Admission Protocol (NAP)

The Network Admission Protocol is used to:

- Register a node in the network and obtain a unique device identifier.
- Provide a way for a node to indicate that it leaves the domain.
- Force a node to leave a domain.

The protocol also performs periodic re-registrations to keep track of the nodes that are present in the domain. The allotted time is configurable and the DM decides its value and announces it in the MAP.

The Spirit HN implements the NAP as specified in the *ITU-T G.9961 Standard, Corrigendum 4, sub clause 8.6.1*.

The process of accepting a node into a G.hn domain, where a DM already sends MAPs, involves the request (`ADM_NodeRegistrRequest.req`) from an EP to the DM to enter the domain. The DM receives this request and evaluates if the domain is ready to accept nodes, and it also performs any additional checks (authentication, MAC filter, etc.) to assess whether the EP can be allowed in.

If the node is authorized, the DM returns a response message (`ADM_DmRegistrResponse.cnf`) to the EP, informing that the node is accepted and providing a device ID (DID) number.

If the node is not authorized, the response message contains a code that indicates why it is not accepted.

Since the nodes have not yet started any channel estimation process, the messages are transmitted in robust communication mode (RCM) for more robustness.

An EP always tries to register with the DM which has the same domain name as the one the EP has been configured with, unless it is in a pairing operation, where multiple candidates to become the node's DM can be available and one can accept the EP in the domain.

In case of rejection, an EP can retry the registration request with the same DM or try a different one if suitable (for example, in a pairing operation where there can be several candidates to become DM).

Once a node is accepted into a domain, that is, it is registered, the EP must renew the registration periodically, by sending a re-registration request message (`ADM_NodeReRegistrRequest.req`) to the DM. The DM renews the registration with this node and sends a response message (`ADM_DmReRegistrResponse.cnf`).

The following figure shows the message sequence for a node that registers in a domain.

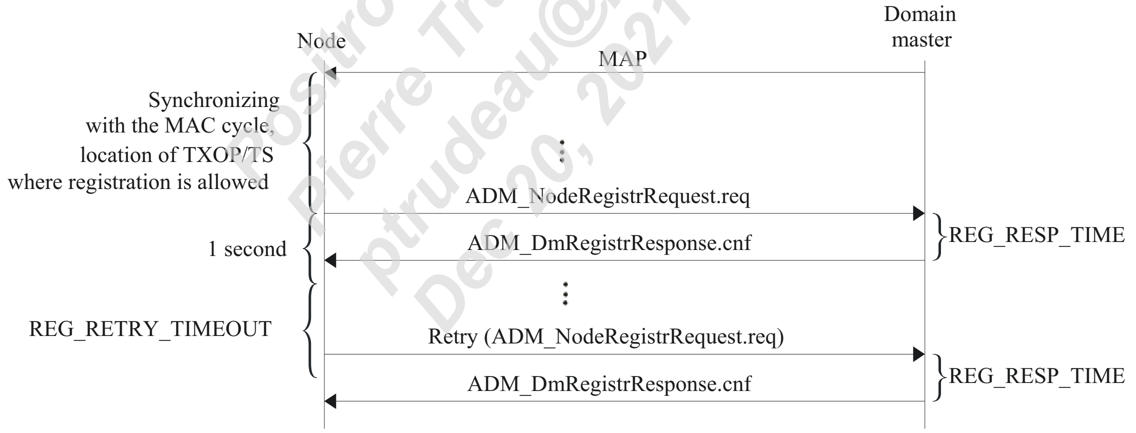

Node
Domain master
MAP
Synchronizing with the MAC cycle, location of TXOP/TS where registration is allowed
ADM_NodeRegistrRequest.req
ADM_DmRegistrResponse.cnf
REG_RESP_TIME
1 second
REG_RETRY_TIMEOUT
Retry (ADM_NodeRegistrRequest.req)
ADM_DmRegistrResponse.cnf
REG_RESP_TIME

Figure 25: NAP Registration Message Sequence

The DM communicates the re-registration period to all nodes in the MAP.

The registration requests require a special slot in the MAP, because there is no DID assigned to the EP that requests to enter the domain. This slot is called registration contention-based time slot (RCBTS) and allows the node that enters the domain to send the registration request messages. The DM sends periodically this slot. By default, there is one every 10 MAPs. The frequency can be modified, making registration faster, but reducing channel efficiency if it is increased.

Once a node is registered, it can decide to leave the domain in a controlled manner, by sending a message indicating so (`ADM_NodeResignRequest.req`). The DM confirms or rejects the request with `ADM_DmResign.cnf`. A DM can also initiate the resignation of a node from the domain with the `ADM_DmForcedResign.req` message.

The following figure shows the message sequence for a node that resigns from a domain.

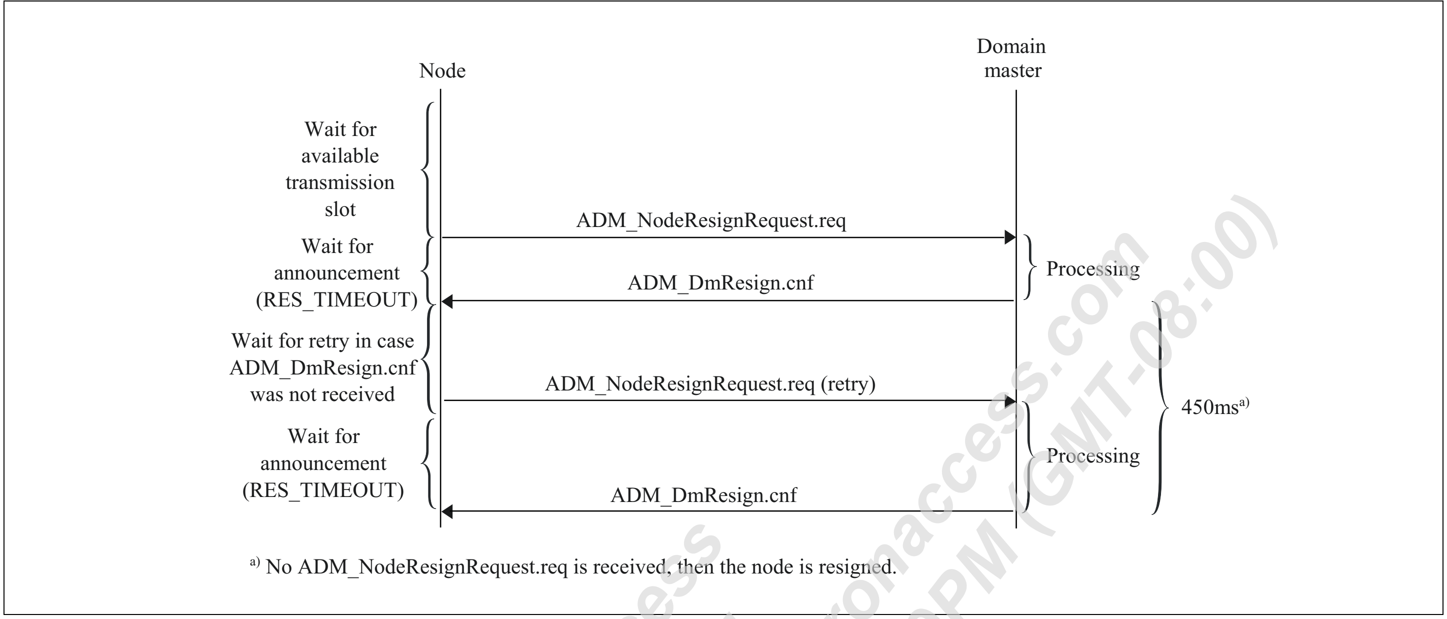

Node
Domain master
Wait for available transmission slot
ADM_NodeResignRequest.req
Processing
Wait for announcement (RES_TIMEOUT)
ADM_DmResign.cnf
Wait for retry in case ADM_DmResign.cnf was not received
ADM_NodeResignRequest.req (retry)
450ms[a)]
Wait for announcement (RES_TIMEOUT)
Processing
ADM_DmResign.cnf
a) No ADM_NodeResignRequest.req is received, then the node is resigned.

Figure 26: NAP Resignation Message Sequence

The Network Admission Protocol messages are also used in the ITU-defined pairing for request and admission of the nodes into a secure domain. They also include versioning information to assess the capabilities of the nodes even before they enter the domain.

Connection Management Protocol

The Spirit HN implements full support of the Connection Management Protocol as specified in the *ITU-T G.9961 Standard (sub clause 8.12).* The Spirit HN software supports data and management connections.

This protocol is used before the exchange of information between two nodes, with the exception of the MAPs and certain messages used in the Network Admission Protocol, which use a special connection of type *connectionless*. The purpose of establishing the connection is to allocate the necessary resources to the associated data flows and to assign properties to it.

The protocol defines the procedure necessary to establish *connections* between a pair of nodes (unicast connections) or between one node and the rest of nodes (broadcast connections).

The connections are unidirectional and can be:

- Tx connections, associated with a hardware set of queues (SOQ).
- Rx connections, associated with a hardware Rx queue.

The connections can be *data* or *management* and they can be configured *with ack* or *without ack*. Each connection can be assigned a certain priority level, depending on the quality of service (QoS) requirements for it.

The Tx node establishes the connections and the Tx or the Rx node can release them. The process of establishing connections involves the exchange of messages between nodes. As the message exchange progresses, the state of each connection is updated to reflect the situation, from *INITIALIZING*, to *CONNECTED* and finally *SOLVED*.

The following table lists the default connections established between two nodes in the Spirit HN.

Table 34: Default Connections Between Two Nodes

Description	Internal Type	Use
Unicast Tx data connection.	`TX_MULTICLASS`	Unicast data transmitted to the other node.
Unicast Tx management connection.	`TX_MGMT`	Unicast management transmitted to the other node.
Broadcast Tx connection.	`TX_BCAST`	Broadcast data and management for all nodes.
Auxiliary internal connection for broadcast and connectionless connection for Tx.	`TX_MGMT_DUMMY`	Required for internal use.
Connectionless connection for Tx.	`TX_CONNLESS`	Only for registration request and response.
Connection for Rx data.	`RX_DATA`	Unicast data received from the other node.
Connection for Rx management.	`RX_MGMT`	Unicast management data received from the other node.
Connection for common channel broadcast reception.	`RX_NDIM_BC`	Management received from the common channel (can be used to communicate with other domains).
Connection used for the reception of the initial registration messages.	`RX_CONNLESS`	Reception of registration request and confirmation messages.

If necessary, there can be more connections established even between the same nodes, depending on the traffic needs and QoS.

If necessary, a connection can be reset, which can only be initiated from the Tx side. To resetting a connection, all pending data in the queues must be flushed and the corresponding segment sequence number (SSN) restarted.

For more information, refer to the *ITU-T G.9960 Standard*.

Routing and Topology Maintenance (RTM) Protocol

The Routing and Topology Maintenance Protocol deals with the maintenance of the routing tables of the nodes in the network. It defines the mechanisms to announce the MAC addresses connected to the Ethernet interfaces of each node in the network. It also describes how to calculate and exchange *unicast routing tables* and *broadcast routing tables*.

The Spirit HN implements the centralized routing and topology management (CRTM) defined by the *ITU-T G.9961 Standard*.

This protocol ensures that all nodes have up-to-date visibility information about the rest of the nodes in the domain. This visibility information includes:

- The MAC addresses that each node sees directly either internally or through its Ethernet interfaces.
- The nodes with direct visibility.
- The route to follow when data must be sent to a certain G.hn node or MAC (or broadcasted).

In an ideal situation, the G.hn nodes have direct visibility of the rest of the nodes in the domain. Since Spirit HN 7.12, the system can also manage situations where EP nodes may not have direct visibility to each other, as long as they have direct visibility with the DM. Traffic between two nodes that have no visibility between them is routed through the DM.

In the RTM protocol mechanism, each node keeps track of the visibility with the rest of the nodes and the MAC addresses it discovers. It populates its local tables with this information and notifies the DM with a message that includes the information. The Spirit HN does not use partial information messages to report this. It always includes all the information. This message is called `TM_NodeTopologyChange.ind`.

When the DM receives a `TM_NodeTopologyChange.ind` message, it updates the local information and sends a broadcast message with the updated data to inform all nodes in the domain of the new situation. This message is called `TM_DomainRoutingChange.ind` and it contains all the information required by any node to manage the routing of any packet within the domain.

There are mechanisms to ensure that all nodes are synchronized in terms of topology information (by broadcasting the last valid sequence number of the `TM_DomainRoutingChange.ind` message in the MAP). This message is sent periodically, but any node can request it to be resent at any time.

The RTM component implements a state machine with events that can be generated locally in the node or by external messages received. These events indicate any change in visibility with other nodes. It can be a new MAC address that is learned from the Ethernet interface, a new DID seen in a G.hn frame PHY header, a node that has left the domain, etc.

All of them force the local tables to be updated and a `TM_NodeTopologyChange.ind` message to be sent to inform the DM of these changes, triggering the process described previously to inform all nodes in the domain.

The RTM component takes care of the following tasks:

- It implements the G.hn communication protocol defined by the ITU to inform all nodes in the domain of any change in visibility or routing between nodes.
- It updates the DID manager tables, which is a database with relevant information about the local and the rest of the nodes in the domain (such as DID, logical port, MAC address, etc.)
- It acts as a driver to populate the bridge forwarding tables (BFTs). The hardware uses these tables to know how to route the traffic. The following figure shows a basic block diagram of the BFTs.

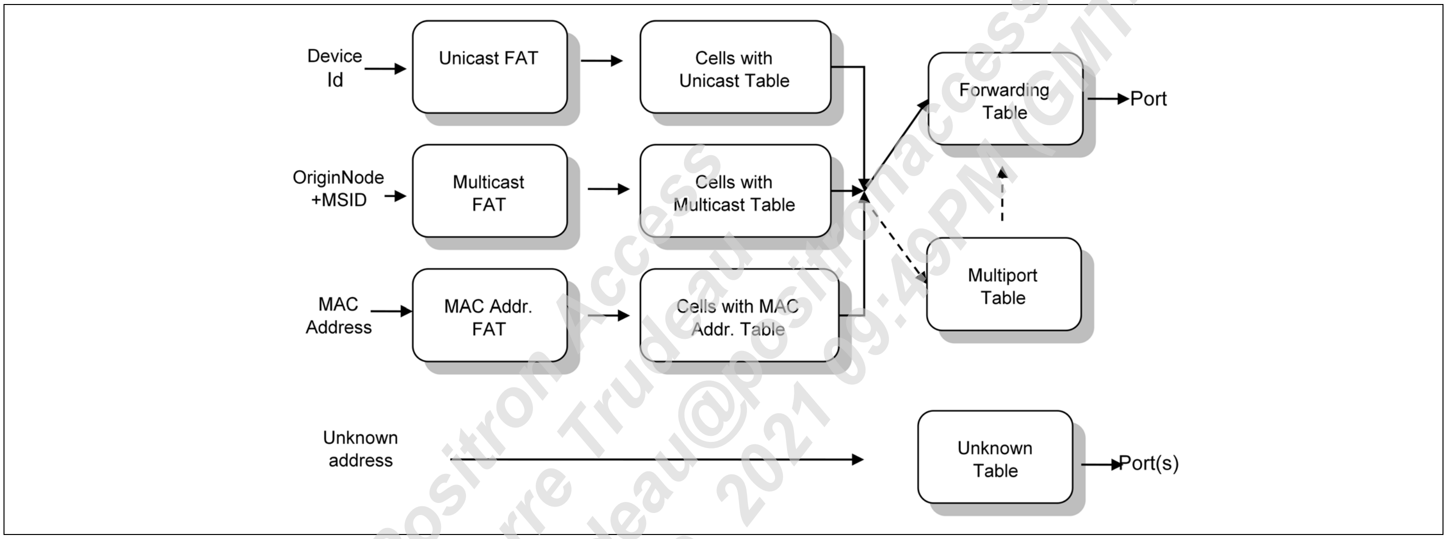


Figure 27: BFTs Overview

Depending on the origin and destination information of the frames to route, a different table is used. There are fast access tables (FATs) that point to *cells* that include the actual addresses or DIDs to use. These cells point to entries in the multiport (in the case of multicast addresses) or the forwarding table, which contains the port information used to send the data. The ports can be, for example, the firmware itself, an Ethernet port, or G.hn ports.

The unknown tables have a different structure and are used to know to which ports a node must replicate data with an unknown destination address.

Broadcast Suppression

This feature limits the broadcast input data rate in the Ethernet port to a maximum value by dropping exceeding packets received.

Table 35: Broadcast Suppression Parameter

Parameter	Description
`CLDRIVER.BCASTSUP.XPUT`	Maximum throughput allowed without suppressing broadcast traffic.The accuracy of this parameter depends on the size of the packets: the bigger the packet, the greater the accuracy. The value 0 disables this feature.

Unknown Traffic Suppression

This feature drops all input traffic, both unicast and multicast, in Ethernet ports whose destination MAC address is not solved, instead of replicating it to all ports.

This feature is disabled by default in the Spirit HN software because the best thing to do with unknown unicast traffic is to broadcast it.

Table 36: Unknown Unicast Traffic Suppression Parameter

Parameter	Description
`RTM.GENERAL.UNKNOWN_SUPPRESSION`	Enables or disables the UNKNOWN destination traffic suppression. **Important:** Do not change this in runtime and make sure that `MCAST.GENERAL.MCAST_FILTERING_ENABLE` = NO to apply this configuration.

Multicast traffic suppression is enabled by default in the HN. Unsubscribed multicast addresses not included in exceptions are dropped to prevent network flooding.

Table 37: Unknown Multicast Traffic Suppression Parameter

Parameter	Description
`MCAST.GENERAL.MCAST_FILTERING_ENABLE`	Enables the multicast filtering feature. If enabled, all unsolicited multicast traffic is blocked. In IPv4 multicast traffic, only traffic between the IP ranges defined in the `MCAST.GENERAL.IGMP_IP_RANGES_DEF` parameter is blocked if unsolicited. **Important:** This feature involves a higher CPU load, so MaxLinear recommends that you enable it only in the video source. Only 100Kbps of broadcast traffic can be managed in this mode.

Traffic and Link Statistics

The Spirit HN software provides a complete set of traffic statistics.

Table 38: Traffic Statistics Parameters

Parameter	Description
`QOS.STATS.G9962_DESC`	*ITU-T G.9962* counters description.
`QOS.STATS.G9962`	*ITU-T G.9962* counters.
`QOS.STATS.RESET`	Global reset for the *ITU-T G.9962* counters.
`ETHIFDRIVER.STATS.ERRORS[18]`	Ethernet Rx and Tx detailed errors statistics. For more information about the columns used in this parameter, see the description for the `ETHIFDRIVER.STATS.ERRORS_DESC` parameter in this table.

Table 38: Traffic Statistics Parameters (Continued)

Parameter	Description
`ETHIFDRIVER.STATS.ERRORS_DESC[351]`	Gets the column description for the `ETHIFDRIVER.STATS.INFO` parameter. ■ TIMER: Timestamp. Free running counter that works with the system clock (100MHz). 32 bits. ■ MSECS: Timestamp. Milliseconds since the last boot. ■ ETHA Tx packets retried due to collision errors. ■ ETHA Tx packets dropped due to maximum collision errors reached. ■ ETHA Rx packets dropped due to reception errors. ■ ETHA Rx packets dropped due to collision errors. ■ ETHA Rx packets dropped due to length errors. ■ ETHA Rx packets dropped due to storage errors (no cells available). ■ ETHA Rx packets dropped due to errors in the data FIFO management. ■ ETHA Rx packets dropped due to cyclic redundancy check (CRC) errors. ■ ETHB Tx packets retried due to collision errors. ■ ETHB Tx packets dropped due to maximum collision errors reached. ■ ETHB Rx packets dropped due to reception errors. ■ ETHB Rx packets dropped due to collision errors. ■ ETHB Rx packets dropped due to length errors. ■ ETHB Rx packets dropped due to storage errors (no cells available). ■ ETHB Rx packets dropped due to errors in the data FIFO management. ■ ETHB Rx packets dropped due to CRC errors.
`ETHIFDRIVER.STATS.INFO[16]`	Ethernet Rx and Tx bytes and packets statistics. For more information about the columns used in this parameter, see the description for the `ETHIFDRIVER.STATS.INFO_DESC` parameter in this table.
`ETHIFDRIVER.STATS.INFO_DESC[247]`	Gets the column description for the `ETHIFDRIVER.STATS.INFO` parameter. ■ TIMER: Timestamp. Free running counter that works with the system clock (100MHz). 32 bits. ■ MSECS: Timestamp. Milliseconds since the last boot. ■ ETHA Tx bytes: Bytes transmitted through the Ethernet interface A (64-bit value). ■ ETHA Rx bytes: Bytes received through the Ethernet interface A (64-bit value). ■ ETHB Tx bytes: Bytes transmitted through the Ethernet interface B (64-bit value). ■ ETHB Rx bytes: Bytes received through the Ethernet interface B (64-bit value). ■ ETHA Tx packets: Packets transmitted through the Ethernet interface A (64-bit value). ■ ETHA Rx packets: Packets received through the Ethernet interface A (64-bit value). ■ ETHB Tx packets: Packets transmitted through the Ethernet interface B (64-bit value). ■ ETHB Rx packets: Packets received through the Ethernet interface B (64-bit value). ■ ETHA Tx Queue Full: Packets discarded due to an Ethernet A buffer full (16-bit value). ■ ETHB Tx Queue Full: Packets discarded due to an Ethernet B buffer full (16-bit value). ■ ETHA Tx total packets with any type of error. ■ ETHA Rx total packets with any type of error. ■ ETHB Tx total packets with any type of error. ■ ETHB Rx total packets with any type of error.

Table 38: Traffic Statistics Parameters (Continued)

Parameter	Description
ETHIFDRIVER.STATS.RESET	When set to *YES*, it resets all statistic counters in the ETHIFDRIVER.STATS.INFO parameter.
FLOWMONITOR.INFO.TX_XPUT_INDICATOR	Estimated application layer Tx throughput. It is an overall estimation, not for a specific link.
FLOWMONITOR.INFO.XPUT_DID	Selected device ID (DID) for the Xput indicator reference. Only valid for FLOWMONITOR.INFO.XPUT_MODE 2, in any other case it returns 0.
FLOWMONITOR.INFO.XPUT_INDICATOR	Estimated application layer Rx throughput. It is an overall estimation, not for a specific link.
FLOWMONITOR.INFO.XPUT_MODE	Mode which allows to select the preferred link used to base the throughput estimation: ■ 1: Uses the node with more traffic as a reference. ■ 2: Uses as a reference its parent node in case of end point and the end point with the highest traffic in case of AM (Only valid in NExt mode).
FLOWMONITOR.STATS.FEC_HISTOGRAM	Gets the column description for the FLOWMONITOR.STATS.FEC_HISTOGRAM parameter. Flow monitor accumulated forward error correction (FEC) iterations histogram per link: **Example:** TIMER,MSECS,SID,TYP,ITER_0%,ITER_1%,ITER_2%,ITER_3%,ITER_4%,ITER_5%,ITER_6%,ITER_7%,ITER_8%,ITER_9%,ITER_10%,ITER_11%,ITER_12%,ITER_13%,ITER_14%. ■ TIMER: Timestamp. Free running counter that works with the system clock (100MHz). 32 bits. ■ MSECS: Timestamp. Milliseconds since the last boot. ■ SID: Source ID. All of the following counters are information related to this device as a source of traffic. ■ TYP: Typical FEC iteration used on the received LLC protocol data units (LPDUs). ■ ITER_X%: Percentage of the LPDUs solved in the FEC iteration X. The final string depends on the product, so it is generated in runtime.
FLOWMONITOR.STATS.FEC_HISTOGRAM_DESC	Flow monitor accumulated statistics per link. For more information about the columns used in this parameter, see the description for the FLOWMONITOR.STATS.FEC_HISTOGRAM parameter in this table.
FLOWMONITOR.STATS.LINK_STATUS[7]	Gets the column description for the FLOWMONITOR.STATS.LINK_STATUS parameter. Flow monitor accumulated statistics per link: ■ TIMER: Timestamp. Free running counter that works with the system clock (100MHz). 32 bits. ■ MSECS: Timestamp. Milliseconds since the last boot. ■ SID: Source ID. All of the following counters are information related to this device as a source of traffic. ■ FRAMES: Number of frames received. ■ LPDUS: Total number of LPDUs received, including valid and errors. ■ ERROR%: Percentage of LPDUs with errors. ■ ABORT%: Percentage of LPDUs aborted.
FLOWMONITOR.STATS.LINK_STATUS_DESC[43]	Flow monitor accumulated statistics per link. For more information about the columns used in this parameter, see the description for the FLOWMONITOR.STATS.LINK_STATUS[7] parameter in this table.

Table 38: Traffic Statistics Parameters (Continued)

Parameter	Description
FLOWMONITOR.STATS.NETSTATS_CONFIG[24]	Arrays of configuration values and thresholds to use in the network statistics for legacy PLC coexistence, MAC scheduling power saving and others: ■ INC_FILT_GAIN_POS: Value added to the increment filter when an increment of Xput is detected. ■ INC_FILT_GAIN_NEG: Value subtracted from the increment filter when an increment of Xput is not detected. ■ DEC_FILT_GAIN_POS: Value added to the decrement filter when a decrement of Xput is detected. ■ DEC_FILT_GAIN_NEG: Value subtracted from the decrement filter when a decrement of Xput is not detected. ■ INC_THR: Threshold value for the increment filter to decide on a real enlargement of the available time slot. ■ DEC_THR: Threshold value for the decrement filter to decide on a real reduction of the available time slot. ■ BRURQ_INC_UDP_THR: Percentage of frames with brurq ! = 0 (which means that the frame has not emptied the transmitter buffer), to declare increment conditions under UDP traffic conditions. ■ BRURQ_INC_TCP_THR: Percentage of frames with brurq ! = 0 (which means that the frame has not emptied the transmitter buffer), to declare increment conditions under TCP traffic conditions. ■ BRURQ_DEC_THR: Percentage of frames with brurq ! = 0 (which means that the frame has not emptied the transmitter buffer), to declare decrement conditions under UDP traffic conditions (with DUR_RATIO_THR). ■ DUR_RATIO_THR: Percentage of the average frame duration in the first part of the MAP cycle, compared to the second part of the MAP cycle. If it is below 100%, it means that the frames in the second part of the MAP cycle are smaller, therefore a reduction is possible. ■ USAGE_THR: Percentage of the duration of frames compared to the total duration available. The last eight MAP cycles are evaluated and if a single MAP cycle does not reach this threshold, the need for a slot increment is considered false. ■ SCHED_INC_GAP: Quantity of increment to apply in the time slot when the conditions are met. ■ SCHED_DEC_GAP: Quantity of decrement to apply in the time slot when the conditions are met. ■ TCP_INC: Quantity of increment to apply in the time slot when the conditions are met under TCP conditions. ■ UDP_MIN_NUM_TX: Minimum number of frames detected in a particular flow in a MAP cycle to consider them for the decision (to avoid spurious traffic altering the decision).
FLOWMONITOR.STATS.NETSTATS_CONFIG_DESC	List of parameters to configure in FLOWMONITOR.STATS.NETSTATS_CONFIG.
FLOWMONITOR.STATS.RESET	When set to *YES,* it resets all statistic counters in the FLOWMONITOR.STATS subgroup.
QOS.STATS.CHANNEL_INFO[10]	Channel adaptation information.
QOS.STATS.CHANNEL_INFO_DESC[70]	Statistics related to the channel adaptation: ■ First Time: Counters for estimations made for the first time. ■ BLER Incr: Number of times the BLER increases. ■ SNR Decr: Number of times the SNR decreases. ■ SNR Incr: Number of times the SNR increases. ■ Adapt End KO: Number of times the adaptation process fails. ■ Adapt End OK: Number of times the adaptation process succeeds.

Table 38: Traffic Statistics Parameters (Continued)

Parameter	Description
`QOS.STATS.DESC[113]`	Gets the column description for the `QOS.STATS.INFO` parameter.
`QOS.STATS.INFO`	QoS statistics: ■ Timer value. ■ Time in milliseconds. ■ Total number of transmitted packets discarded. ■ Total number of packets transmitted. ■ List of DIDs for which the node discards transmitted packets. Note: The size of the parameter depends on the DIDs supported by the product, so the array size in the type column is not exact.
`QOS.STATS.DEVID_TRAFF_CLASS`	Device ID to monitor the discarded traffic class queue.
`QOS.STATS.G9962[23]`	Throughput statistics information. All of the elements described in this section are free-running counters. These counters are reset when the status changes to *Up* after the node is enabled by the node management entity (NME).
`QOS.STATS.G9962_DESC[295]`	`QOS_STATS_G9962` elements description: `BytesSent, BytesReceived, PacketsSent, PacketsReceived, ErrorsSent, ErrorsReceived, UnicastPacketsSent, UnicastPacketsReceived, DiscardPacketsSent, DiscardPacketsReceived, MulticastPacketsSent, MulticastPacketsReceived, BroadcastPacketsSent, BroadcastPacketsReceived, UnknownProtoPacketsReceived, MgmtBytesSent, MgmtBytesReceived, MgmtBytesPacketsSent, MgmtBytesPacketsReceived, BlocksSent, BlocksReceived, BlocksRsent, BlocksErrorReceived.`
`QOS.STATS.RESET`	Resets all the statistic counters.
`QOS.STATS.RX_LLC_ERRORS[2]`	Number of erroneous logical link controls (LLCs) received. The counters are classified by error cause.
`QOS.STATS.RX_LLC_ERRORS_DESC[38]`	LLCs received with wrong CRC, LLCs received with wrong cipher MIC.
`QOS.STATS.TCLASS_DISC_TX_QS[5]`	Traffic class Tx queues that discard packets.
`QOS.STATS.TCLASS_DISC_TX_QS_DESC[179]`	For each remote node with which there is a PLC connection, this parameter shows: ■ Device ID of the remote node to which it transmits. ■ Tx queues that discard packets in priority (1,2). ■ Tx queues that discard packets in priority (0,3). ■ Tx queues that discard packets in priority (4,5). ■ Tx queues that discard packets in priority (6,7).
`QOS.STATS.TRAFF_CLASS_DISC[4]`	Traffic class discarded Tx packet counters for the specified device ID.
`QOS.STATS.TRAFF_CLASS_DISC_DESC[144]`	■ Tx packets discarded in priority (1,2). ■ Tx packets discarded in priority (0,3). ■ Tx packets discarded in priority (4,5). ■ Tx packets discarded in priority (6,7).

Table 38: Traffic Statistics Parameters (Continued)

Parameter	Description
`CHEST.INFO.TX_DESC`	Gets the column description of the `CHEST.INFO.TX` parameter. The final number of columns depends on the number of supported regions. ■ `SnID`: Session node ID. All the information of the selected row is related to this SnID. ■ `BPS[region]`: BPS for this SnID and this region. ■ `FECR[region]`: FEC rate for this SnID and this region. ■ `RCM[region]`: Indicates if the robust communication mode (RCM) is used in this SnID and this region. ■ `nREP[region]`: Number of repetitions used in this SnID and this region. Only when using the RCM. Here is an example of columns when using two regions: SnID,BPS0,FECR0,RCM0,nREP0,BPS1,FECR1,RCM1,nREP1
`CHEST.INFO.TX`	Channel status information. For more information about the columns used in this parameter, see the description for the `CHEST.INFO.TX_DESC` parameter in this table.

Traffic Shaping

The traffic shaping conforms the traffic received from the G.hn interface and transmits it to the Ethernet port at the same rate as it was initially received. The bursty nature of the G.hn transmissions does not overload the interfaces of external devices which can cause packet loss.

The traffic shaping is important for HN devices connected to low speed devices with 100Mbps Ethernet interfaces because they can receive data on their G.hn interfaces at up to 1.6Gbps.

The following table lists the configuration parameters that enable the traffic shaping.

Table 39: Traffic Shaping Parameters

Parameter	Description
`FLOWMONITOR.TRAFFSHAP.CURRENT_LIMIT`	Current throughput limit applied (only valid when the traffic shaping feature is enabled).
`FLOWMONITOR.TRAFFSHAP.ENABLED`	Enables or disables the use of the traffic shaping feature.
`FLOWMONITOR.TRAFFSHAP.FILTER[2]`	The estimated ingress throughput is calculated as follows: (prev_xput * (100% - filter)) + (new_xput * filter). There are two values: ■ The first one is used when the traffic rate increases. ■ The second one is used when the traffic rate decreases.
`FLOWMONITOR.TRAFFSHAP.FORCED_RATE`	Forces the traffic shaping rate. When this feature is enabled, no ingress rate detection algorithm is used, and you directly force the configured rate. The value 0 disables this feature and tries to automatically detect the ingress rate. `TRAFFSHAP` must be enabled to be able to force the rate.
`FLOWMONITOR.TRAFFSHAP.GUARD`	The throughput limit is calculated as follows: estimated_xput + (estimated_xput * guard).

Table 39: Traffic Shaping Parameters (Continued)

Parameter	Description

`FLOWMONITOR.TRAFFSHAP.INTERNAL100M`	Enables or disables an accurate 100Mbps shaping only for the MACs forwarded to the configured interface.
	When this feature is enabled, you must set `FLOWCONTROL.ETH100.ENABLED` to *NO* to avoid conflicts.
	The MACs connected to an external switch port at 100Mbps must be forwarded to a particular interface. To forward the MACs to the INTERNAL100M interface, use the `BFT.GENERAL.MAC_ADD` parameter, taking into account that ETHA = port 2 and ETHB = port 3.
	For example, in a product with an external switch connected to ETHA, you must perform the following configuration:
	`ETHIFDRIVER.ETHB.INTERNAL_PHY = YES`
	`ETHIFDRIVER.ETHB.ENABLED = YES`
	`FLOWCONTROL.ETH100.ENABLED = NO`
	`FLOWMONITOR.TRAFFSHAP.INTERNAL100M = 2`
	To perform accurate shaping for MAC 00:11:22:33:44:55, use:
	`BFT.GENERAL.MAC_ADD` = 0x00,0x11,0x22,0x33,0x44,0x55,0x3,0,0

`FLOWMONITOR.TRAFFSHAP.LIMIT_RX_BUFF_ETH100_EN`	Limits the maximum amount of data per Tx when Ethernet is connected at 100Mbps. When this feature is enabled, if Ethernet is connected at a speed lower than 1Gbps, the maximum number of FEC blocks per transmission is limited to the `FLOWMONITOR_TRAFFSHAP_MAX_RX_BUFFER_ETH100` value.
`FLOWMONITOR.TRAFFSHAP.MAX_RX_BUFFER_ETH100`	Maximum number of FEC blocks allowed to transmit per Tx when Ethernet is connected at 100Mbps.
`FLOWMONITOR.TRAFFSHAP.WINSIZE`	Configures the window size used to measure the ingress throughput.

VLAN

The *IEEE 802.1D* and *IEEE 802.1Q* standards define a system of virtual LAN (VLAN) tagging for Ethernet frames by adding a header to Ethernet frames and specifying its use by bridges and switches.

The Spirit HN software partially supports the *IEEE 802.1Q* by implementing the tagging/untagging and filtering capabilities on the Ethernet and management interfaces.

Tagging

The system can add a configured VLAN tag to incoming untagged Ethernet frames and can remove this particular VLAN tag to outgoing Ethernet frames by leaving unchanged the rest of VLAN tags. The same behavior is applied to management interface.

The tagging can be defined on each interface as:

- ACCESS: Ingress packets are tagged with the specified VLAN tag for this interface and for egress packets, the VLAN tag is removed.
- TRUNK: When a VLAN tag is specified as the port VLAN ID (PVID) in a trunk interface, the ingress packets without a VLAN tag are tagged with the specified VLAN tag, and for egress packets with this VLAN tag, the VLAN tag is removed. Otherwise, the VLAN tag is unchanged. If the PVID is not specified in a trunk interface (PVID is zero), the packets are not modified.
- NONE: Packets are not modified.

Filtering

The Spirit HN software implements a VLAN filtering feature based on the available set of VLAN tags and is applied in ingress and egress on each interface.

The set of tags can be defined independently per each interface and the maximum number of the available VLAN tags is 16.

The filtering on each interface depends on its type and is defined as:

- ACCESS: On this type of interface, only the VLAN tag specified for the interface is available, so packets with a VLAN tag other than that specified for the interface are dropped.
- TRUNK: Packets with VLAN tag are available if the tag is in the tag list.

Table 40: VLAN Configuration Parameters

Parameter	Description
`VLAN.CVLAN.ENABLE`	Activates or deactivates the VLAN (*IEEE 802.1Q*).
`VLAN.CVLAN.FILTERING_ENABLE`	Enables or disables the VLAN ingress and egress filtering.
`VLAN.CVLAN.PVID_ETHA`	VLAN identifier for the Ethernet A interface. If set to *0*, tagging is deactivated.
`VLAN.CVLAN.PVID_ETHB`	VLAN identifier for the Ethernet B interface. If set to *0*, tagging is deactivated.
`VLAN.CVLAN.PVID_MGMT`	VLAN identifier for the management interface. If set to *0*, tagging is deactivated.
`VLAN.CVLAN.PVID_SDIO`	VLAN identifier for the SDIO interface. If set to *0*, tagging is deactivated.
`VLAN.CVLAN.CONFIG_IF_ETHA`	Port configuration for the Ethernet A interface. The values are: ■ ACCESS ■ TRUNK ■ NONE
`VLAN.CVLAN.CONFIG_IF_ETHB`	Port configuration for the Ethernet B interface. The values are: ■ ACCESS ■ TRUNK ■ NONE
`VLAN.CVLAN.CONFIG_IF_MGMT`	Port configuration for the management interface. The values are: ■ ACCESS ■ TRUNK ■ NONE
`VLAN.CVLAN.CONFIG_IF_SDIO`	Port configuration for the SDIO interface. The values are: ■ ACCESS ■ TRUNK ■ NONE
`VLAN.CVLAN.ALLOWED_TAGS_IN_ETHA`	Tags allowed on the Ethernet A interface.
`VLAN.CVLAN.ALLOWED_TAGS_IN_ETHB`	Tags allowed on the Ethernet B interface.
`VLAN.CVLAN.ALLOWED_TAGS_IN_FW`	Tags allowed on the management interface.
`VLAN.CVLAN.ALLOWED_TAGS_IN_SDIO`	Tags allowed on the SDIO interface.

Transparent Mode with Management VLAN

The system can configure a transparent VLAN mode for data traffic while using a management VLAN to access modems configuration and management.

The configuration is as follows:

```
VLAN.CVLAN.ENABLE = YES
VLAN.CVLAN.FILTERING_ENABLE = YES
VLAN.CVLAN.PVID_ETHA = 0
VLAN.CVLAN.PVID_ETHB = 0
VLAN.CVLAN.PVID_MGMT = <MANAGEMENT_VLAN>
VLAN.CVLAN.PVID_SDIO = 0
VLAN.CVLAN.CONFIG_IF_ETHA = NONE
VLAN.CVLAN.CONFIG_IF_ETHB = NONE
VLAN.CVLAN.CONFIG_IF_MGMT = ACCESS
VLAN.CVLAN.CONFIG_IF_SDIO = NONE
VLAN.CVLAN.ALLOWED_TAGS_IN_ETHA = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
VLAN.CVLAN.ALLOWED_TAGS_IN_ETHB = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
VLAN.CVLAN.ALLOWED_TAGS_IN_FW = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
VLAN.CVLAN.ALLOWED_TAGS_IN_SDIO = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Traffic Prioritization

The Spirit HN software implements traffic classification and prioritization based on the *IEEE 802.1p* standard, DSCP, and user-defined rules. Prioritization is only applied on incoming traffic via Ethernet at node level, not at network level. For more information, see "Quality of Service (QoS)" on page 36.

The Spirit HN software also implements a predefined set of traffic prioritization for well-known traffic, such as ARP and TCP IPv4/IPv6 acknowledgments.

Prioritization Rules Order

The Spirit HN software allows you to define the order of priority between I*EEE 802.1p* and DSCP classification mechanisms. It there are enabled custom rules and predefined rules for well-known traffic prioritization, they are applied prior to *IEEE 802.1p* and DSCP.

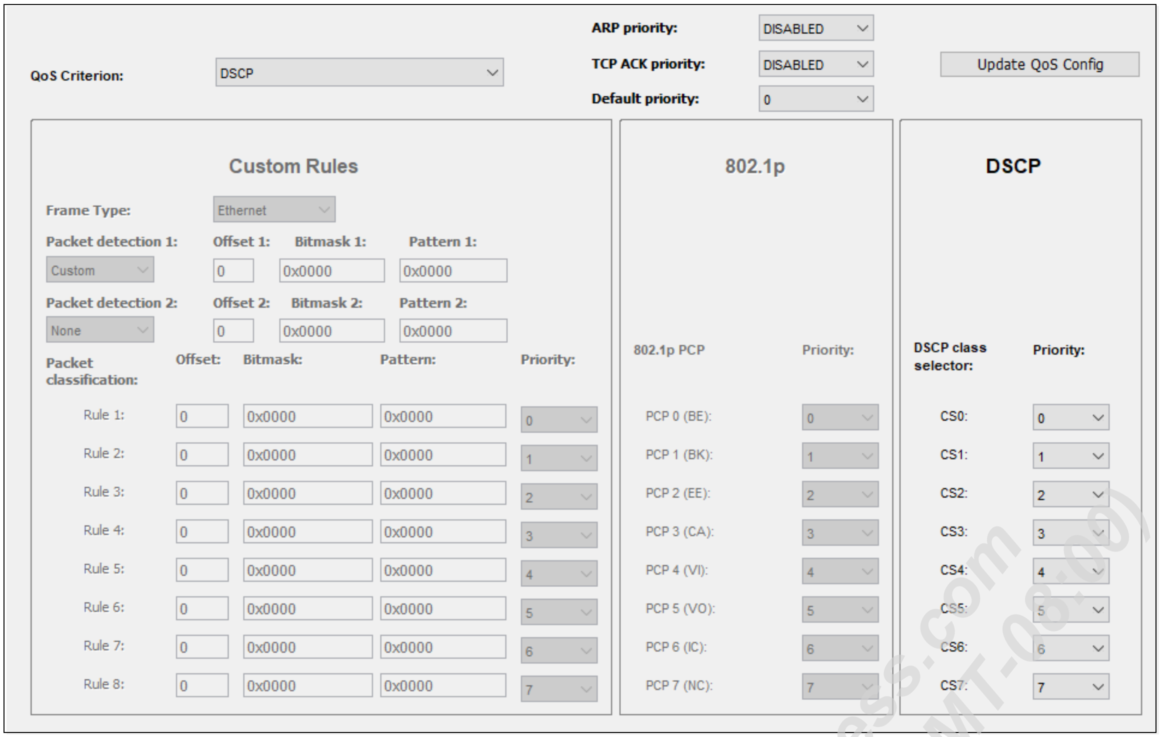
The possible values of the rules order are:

- VLAN: The classification is based only on VLAN information.
- DSCP: The classification is based only on DSCP information.
- VLAN_DSCP: The classification is based on VLAN and DSCP, and the VLAN information prevails if there are any contradictory classification orders.
- DSCP_VLAN: The classification is based on VLAN and DSCP, and the DSCP information prevails if there are any contradictory classification orders.

Table 41: Prioritization Rules Order Configuration

Parameter	Default Value
PACKETCLASSIFIER.GENERAL.RULES_ORDER	DSCP_VLAN

The following figure shows the different prioritization rule options found in the Spirit Configuration Tool (SCT).



QoS Criterion: DSCP

ARP priority: DISABLED

TCP ACK priority: DISABLED

Default priority: 0

Custom Rules

Frame Type: Ethernet

Packet detection 1:	Offset 1:	Bitmask 1:	Pattern 1:
Custom	0	0x0000	0x0000

Packet detection 2:	Offset 2:	Bitmask 2:	Pattern 2:
None	0	0x0000	0x0000

Packet classification:	Offset:	Bitmask:	Pattern:	Priority:
Rule 1:	0	0x0000	0x0000	0
Rule 2:	0	0x0000	0x0000	1
Rule 3:	0	0x0000	0x0000	2
Rule 4:	0	0x0000	0x0000	3
Rule 5:	0	0x0000	0x0000	4
Rule 6:	0	0x0000	0x0000	5
Rule 7:	0	0x0000	0x0000	6
Rule 8:	0	0x0000	0x0000	7

802.1p

802.1p PCP	Priority:
PCP 0 (BE):	0
PCP 1 (BK):	1
PCP 2 (EE):	2
PCP 3 (CA):	3
PCP 4 (VI):	4
PCP 5 (VO):	5
PCP 6 (IC):	6
PCP 7 (NC):	7

DSCP

DSCP class selector:	Priority:
CS0:	0
CS1:	1
CS2:	2
CS3:	3
CS4:	4
CS5:	5
CS6:	6
CS7:	7

Figure 28: Prioritization Rule Options in SCT

IEEE 802.1p Support

The *IEEE 802.1p* standard (part of the *IEEE 802.1D*) defines a set of eight classes of services expressed as the 3-bit PCP field in the *IEEE 802.1Q* header (VLAN header) in the Ethernet frame.

The Spirit HN software implements the mapping between the *IEEE 802.1p* traffic classes and the ITU-T G.hn class.

The following table lists the corresponding configuration parameters.

Table 42: IEEE 802.1p Prioritization Parameters

Parameter	Description
PACKETCLASSIFIER.GENERAL.VLAN_CLASS_MAP_EN	The values are: ■ YES: Enabled. ■ NO (default value): Disabled.
PACKETCLASSIFIER.GENERAL.VLAN_CLASS_MAP	Table where each position represents an *IEEE 802.1p* priority. The value associated to a position is the G.hn class for this 802.1p priority. The default values are: 2,0,1,3,4,5,6,7.

DSCP Support

Differentiated services (DiffServ) is a traffic management model that specifies a mechanism to classify and manage the network traffic and provide the quality of service on the IP networks.

The DiffServ architecture defines the DiffServ (DS) field that supersedes the ToS field in IPv4. Inside the DS field, the six most significant bits represent the differentiated services code point (DSCP). Inside the DSCP, the most significant bits are used to map the precedence, being backwards compatible with the IP precedence bits in the ToS field.

The Spirit HN implements a mapping between the DSCP and the ITU-T G.hn classes.

The following table lists the corresponding parameters.

Table 43: DSCP Configuration

Parameter	Description
`PACKETCLASSIFIER.GENERAL.DSCP_CLASS_MAP`	Table where each position represents a DSCP value. The value associated to a position is the G.hn class for this DSCP value. For example, the DSCP value 8 is associated to the `CLASS_MAP[8]` value (equal to 1), thus the DSCP value 8 is classified as class 1. The default values are: 0,0,0,0,0,0,0,0, 1,1,1,1,1,1,1,1, 2,2,2,2,2,2,2,2, 3,3,3,3,3,3,3,3, 4,4,4,4,4,4,4,4, 5,5,5,5,5,5,5,5, 6,6,6,6,6,6,6,6, 7,7,7,7,7,7,7,7
`PACKETCLASSIFIER.GENERAL.DSCP_CLASS_MAP_EN`	The values are: ■ YES (default value): Enables the DSCP to the G.hn class mapping. ■ NO: Disables the DSCP to the G.hn class mapping.

Custom Rules

The Spirit HN software allows you to configure the user-defined prioritization rules by defining a pattern matching rule and a set of packet classification rules.

There are two pattern matching rules that can be enabled independently.

A packet matching rule is defined by:

- Offset: Offset inside the Ethernet packet where the bitmask and pattern must be applied. The offset is in 16-bit units. Offset 0 is the first byte of an Ethernet packet (destination MAC address LSB).
- Bitmask: 16-bit bitmask to apply to the value (Value) in the Ethernet packet for the specified offset.
- Pattern: 16-bit pattern. If the result of applying the bitmask to the value is equal to the pattern, the rule results is matched.

The matching rules are applied to incoming packets through the Ethernet interface. When there is a match, the classification rules are applied to the packet.

The classification rules are defined similarly to the packet matching rules. It can be defined up to eight rules and each rule can be mapped to a G.hn ITU-T class.

Table 44: Custom Rules Parameters

Parameter	Description
`PACKETCLASSIFIER.GENERAL.TYPE_CLASS_MAP_EN`	General enabler for custom rules. The values are: ■ YES: The defined custom rules are configured in the packet classification hardware. ■ NO (default value): Disabled.
`PACKETCLASSIFIER.GENERAL.TYPE_CLASS_MAP`	Table that contains the mapping between classify rules and priorities.

Well-Known Traffic Prioritization

The Spirit HN software implements a set of predefined rules to enable the prioritization of well-known types of traffic such as:

- TCP/IP ACK frames: Prioritizing TCP packets that carry only ACK frames reduces round-trip time and prevents losses on these frames under congested environments. It increases the TCP traffic performance.
- ARP: Frames that are used to address resolution. Prioritizing these packets guarantees that the ARP protocol continues working under congested environments.

The following table lists the corresponding configuration parameters.

Table 45: TCP ACK Prioritization Parameters

Parameter	Value
`PACKETCLASSIFIER.GENERAL.TCPACKV4_CLASS_MAP_EN`	The values are: - YES (default value): Enabled. - NO: Disabled.
`PACKETCLASSIFIER.GENERAL.TCPACKV4_CLASS_MAP`	Priority level. 0–7 (default: 4)
`PACKETCLASSIFIER.GENERAL.TCPACKV6_CLASS_MAP_EN`	The values are: - YES (default value): Enabled. - NO: Disabled.
`PACKETCLASSIFIER.GENERAL.TCPACKV6_CLASS_MAP`	Priority level. 0–7 (default: 4)
`PACKETCLASSIFIER.GENERAL.ARP_CLASS_MAP_EN`	The values are: - YES (default value): Enabled. - NO: Disabled.
`PACKETCLASSIFIER.GENERAL.ARP_CLASS_MAP`	Priority level. 0–7 (default: 6)

I-temp Chipset Support

The Spirit HN software supports the 88LX5153A and 88LX2741 chipset to work with industrial temperature range (from –40°C to 85°C).

Application Features

This section describes the application layer features included in the Spirit HN software.

User Interface

This section describes the hardware user interface supported in the Spirit HN software.

GPIOs Configuration

The device contains a set of GPIOs to manage the external hardware within the firmware. The provided API allows you to configure, read, write, and trigger the interrupt requests (IRQs) on any of the available GPIOs.

When the device has a button or a LED connected to a GPIO, the Spirit HN software also contains additional APIs to register a button that is pressed or program LED states.

Buttons and LEDs

You can configure buttons and LEDs as required. The API allows you to update the state of the different LEDs connected to the device, depending on a variety of inputs. You can do this mainly through the *connectivity feedback* component. This component adds a layer between the components that generate the data to be *visualized* and the actual activation of the LEDs. LEDs are only a particular method to visualize something. It can be sending a message to an external device or any other action that allows to show something.

The connectivity feedback component also allows the same element, such as a LED, to be driven by information from different sources. The priority of each of these sources determines which is currently used to drive the LED. For example, you can use a LED to show the power status, but during a pairing operation, it can blink to indicate the current activity. When the operation is completed, the LED can revert to show the power status again.

By default, the LEDs are updated with information related to the power, pairing, quality of the link, or the activity of the Ethernet interface and buttons are associated with functionalities such as the factory reset or pairing.

For more information about the user interface, refer to the *G.hn Spirit Firmware Customization Programming Guide* (006PG).

The following table lists the configuration parameters of the user interface.

Table 46: User Interface Configuration Parameters

Parameter	Description
`UI.BUTTON.GPIO_NUMBER[7]`	GPIO number assigned to each button. This assignment is used by other components. For example, if `PAIRING.BUTTON.NUMBER` is set to *0*, the pairing button GPIO is configured in the position 0 of this array. Set the value to *255* to disable the button.
`UI.BUTTON.GPIO_PRESSED_VALUE[7]`	When the specified value is read from the GPIO, the UI component considers that the button is pressed.
`UI.LED.GPIO_NUMBER[12]`	GPIO number assigned to each LED. This assignment is used by other components. For example, if `CONNECTIVITYFB.POWER.LED_NUMBER` is set to *2*, the power LED GPIO is configured in the position 2 of this array. Set the value to *255* to disable the LED.
`UI.LED.GPIO_ON_VALUE[12]`	When indicating to the UI component to *switch on* the LED, this is the value that is written to the GPIO.

Configuration Layer

The Spirit HN software implements a configuration layer that contains all the configuration parameters of the MaxLinear's G.hn node. It is used by other firmware components to configure its operation mode and report information. It also provides a single abstraction layer between firmware components, user interface, and adapter management tools.

To access the configuration parameters, you can use the Spirit Configuration Tool (SCT) or the web server embedded in the system.

There are parameters only for POST operations (write-only), only for GET operations (read-only), or for both POST and GET operations (read-write). To access them, you can use the **Advanced Configuration** tab of the SC Tool for MaxLinear's G.hn devices.

All parameters that need to be persistent after a reset are stored in a file in the flash FS. You can configure this file by using the product configuration kit (PCK). This kit provides a GUI tool to encode the configuration parameters and build the final flash and configuration images that are used in the G.hn devices. For more information about the GUI tool, refer to the help menu in the tool.

The parameters of the configuration layer are structured hierarchically in three levels:

- A first level is called *group* and is usually associated with a specific component or a main feature.
- Groups can be split into one or more subgroups, which is more logical when a complex feature has several aspects that are better managed if they are split into smaller chunks.
- Each subgroup includes individual parameters.

A type is assigned to the configuration layer parameters. Certain are simple types, such as CHAR, INT32U, etc. Others are more complex, such as IP or MAC addresses, and then they can be grouped into arrays or even 2D tables. Customers can create new custom types if required.

Layer 2 Configuration and Management Protocol (LCMP)

The LCMP is defined by the ITU and is primarily conceived as a protocol wrapper to extend the current set of management messages defined for G.hn. By encapsulating messages into LCMP packets, it is possible to implement full protocols between the G.hn devices or between the G.hn devices and external equipment, which was the initial approach for the LCMP.

The LCMP management messages allow the use of the same mechanisms defined for general management messages in G.hn, such as encryption, segmentation, or routing, while extending the possibilities of defining protocols or data models at different levels.

Certain of these *data models,* such as the Certification and Interoperability program from HomeGrid, are agreed upon at higher levels, while others, such as the Maxlinear configuration layer, can be vendor-specific. The LCMP provides mechanisms to determine which data model is used on each LCMP message. Parameters can be sent encrypted if required. Certain parameters of a certain data model can be encrypted while others remain open if required.

The Spirit HN software supports for the LCMP and implements the data model required for the HomeGrid Forum compliance and interoperability (C&I) automation program.

In addition, a private data model is defined exposing the configuration layer. You can access this data model by using the device embedding kit (DEK).

Other data models included in the system can be for example the remote file access, which allows to boot the G.hn firmware without a physical flash—implementing an interface with a file system located in a host machine—, or the CMA data model, for channel monitoring measurements.

Although LCMP was initially defined for communication between G.hn devices and external devices through an Ethernet interface, the LCMP frames are actually G.hn management messages and they can also be sent between the G.hn nodes.

This can be very useful to query information or trigger actions in other nodes in the domain. The Spirit HN software provides an API to send and receive these messages in order to set and get any configuration layer parameter from other nodes in the domain. In the Spirit HN, these messages are called *internode LCMP messages*.

Web Server

The embedded web server allows you a remote management and configuration of the device using a standard web browser.

The static pages are stored in the flash FS under the **B:\web** folder. These pages can be customized and uploaded to the modem at runtime using the one step upgrade procedure (OSUP) feature. You can also use the PCK tool to create customized flash images with custom web content before flashing the devices.

JavaScript is used to access modem information through the configuration layer.

Note: The default web page is an example and it does not allow you to configure all of the available parameters.

Log File

The Spirit HN software implements an application to capture periodically information from the node. The information is stored in a local file that can be uploaded also periodically to a remote FTP server. Therefore, you can remotely monitor the operation of the network, capture the relevant information, and analyze it for problem troubleshooting.

The maximum log file size is 256Kbytes and the files uploaded to the FTP server are named according to the *“logfile_”+”MAC address”+”sequence number”* pattern. You can then easily identify different log files.

Example: logfile_00b9d093ac23_23

This feature also allows you also to monitor any parameter included in the configuration layer. The parameters to monitor are stored in a text file in the flash FS, located in **b:/logfile/logfile.cfg**.

The file can be updated with the one step upgrade procedure (OSUP), using the option to upgrade a file in the file system.

Table 47: Example of Configuration File for Log File Feature

```
SYSTEM.MISC.UPTIME
DIDMNG.GENERAL.DIDS
DIDMNG.GENERAL.MACS
DIDMNG.GENERAL.TX_BPS
DIDMNG.GENERAL.RX_BPS
FLOWMONITOR.STATS.FEC_HISTOGRAMMASTERSELECTION.STATS.DESC
MASTERSELECTION.STATS.INFO
```

You can configure this feature with the log file configuration parameters described in the *Log File Kit User Guide* (051UG).

Table 48: Log File Parameters

Parameter	Description
LOGFILE.GENERAL.DATA_INTERVAL	Time interval between logs (in seconds).
LOGFILE.GENERAL.UPLOAD_INTERVAL	Time interval between automatic uploads of the log file to an external FTP server (minutes). If set to *0*, it disables the automatic upload.
LOGFILE.GENERAL.ENABLE	Specifies if the logging feature is enabled.
LOGFILE.GENERAL.SEND_FILE	When written, the log file is sent to the local disk or to the FTP server. When read, it returns the status of sending file progress: ■ 0: Finished. ■ 1: Still in progress.
LOGFILE.GENERAL.DESTINATION	Configures the log file destination as follows: ■ LOCAL: The log file is stored in the equipment in the **b:/logfile/logfile_last** directory. ■ FTP: The log file is sent by the FTP to the server.
LOGFILE.GENERAL.STATUS	Reports information about the log process (both errors and status).
LOGFILE.FTPSERVER.HOST	URL or IP (IPv4 or IPv6) of the FTP server.
LOGFILE.FTPSERVER.LOGIN	Login to connect to the FTP server.
LOGFILE.FTPSERVER.PASSWORD	Password to connect to the FTP server.

Flash Image

The flash image is the file that represents all the contents of the flash. When the flash memory of the modems is completely overwritten with one of these images, it is as if its *hard drive* is filled with a whole new set of all the files required to boot and operate. After the modem operates, the files inside the image are updated with new information, etc. This section describes the different parts of this image and how it is generated.

The following figure shows the Flash Memory sections.

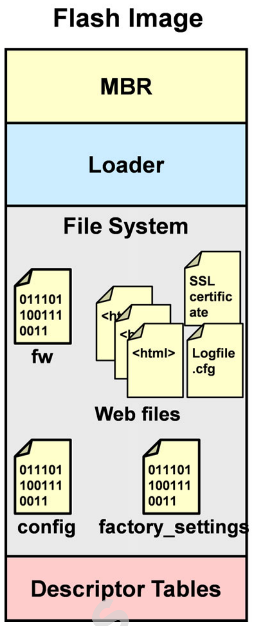


Figure 29: Flash Memory Sections

The upgrade system is built around the flash file system (FFS). The different sections of the file system are files, with the exception of the Master Boot Record (MBR), the Loader, and the Descriptor Tables. The one step upgrade procedure (OSUP) allows the upgrade section in the file system and the Loader. However, the MBR and Descriptor Tables sections cannot be modified with any of the upgrade processes. The firmware only accesses them to manage production-related data and the file system.

The following table lists the main sections or system files in the flash.

Table 49: Sections and Flash File System

File System	System Section	Comment
No	MBR	Read-only data, such as serial number and MAC address.
	Loader	Launches the firmware.
	Descriptor Tables	Used for management of the file system.
Yes	Firmware	Running firmware.
	Config	Current device configuration.
	Factory	Device configuration from factory.
	File	Any other file in the FFS, such as Web and TR-069 files.

Flash Production Section

The flash production is a section in the flash device where configurations that are written during production are stored. The configurations are displayed as write-protected configurations only, to avoid accidental modifications.

The typical parameters stored in this section are:

- Boot pattern.
- MBR Version.
- Loader Offset.
- HW Product.
- HW Revision.
- Product Class.
- Device Manufacturer.
- Device Name.
- Device Description.
- Device Serial Number.
- Device Ethernet MAC Address.
- AFE Calibration information.
- Custom Fields.
- Factory Profile.

You can set these parameters in two different ways:

- Using the product configuration kit (PCK): When generating a complete flash image.
- Using the production test kit (PTK): When executing the production test.

For more information, refer to the help embedded in both tools delivered with the firmware.

Loader Support

The Spirit HN implements a Loader module that initializes the digital baseband processor and any external components required to load the firmware into memory.

The Loader module configures the hardware blocks required to access Double Data Rate Synchronous Dynamic Random-Access Memory (DDR SDRAM)—usually referred as DDR or RAM—, Ethernet ports, etc., by locating the firmware image in the flash file system, reading it and unzipping it from flash to RAM, and launching the firmware execution after the device initialization is complete.

The Loader module is not included in the flash file system. There are two fixed sections in flash to store two images of the Loader module and allow remote upgrade of the Loader in a secure manner (fail-safe).

Boot from Host (Flashless)

This feature allows you to create G.hn products without flash, for example G.hn + WiFi extenders. The host provides and maintains the loader, firmware, and configuration files.

Generating a G.hn flashless product requires a firmware that supports remote file access and the creation of a host software that provides the files to the G.hn node.

The communication between the firmware and the host tool is based on the LCMP protocol (L2 Ethernet frames), so both devices must have Ethernet connectivity.

The firmware used by this product must have the following features:

- Flash dummy view.
- File system in RAM (RFS, RAM file system). The set of files required by the G.hn firmware is stored in the host system. The G.hn RAM file system keeps a copy of this set of files to reduce the overload of file access in the Ethernet interface.
- LCMP for updating remote files using the remote file access (RFA) data model.

A host software called GhnFlashlessService included in this release is necessary to manage all the new actions required by the lack of flash in the G.hn device. It starts a daemon to monitor the device and process the requests from it.
Its main tasks are to:

- Manage the files required by the G.hn flashless device in the local storage.
- Detect when the device boots and requests a firmware to load.
- Prepare the firmware image by embedding in it the files required for the RAM file system.
- Serve the firmware to the device.
- Receive LCMP requests for file operations to maintain consistence between the firmware files copy in RAM and the files in the host storage. Since the G.hn device manages a local copy of the files in its RAM file system, only write, create, and remove actions are applied to the host files.

The Spirit HN host tool only supports one G.hn device. MaxLinear recommends that you limit the Ethernet visibility from the host to only one G.hn flashless device.

A flashless product can include most of the same features as a product with flash except those directly related to the inclusion of a flash component, such as the firmware storage in encrypted or unencrypted flash.

> **Note:** Take into account the possible additional delay due to remote file access when reading or writing configuration parameters or other actions on the remote file system.

The host manages the firmware upgrade, but flashless products can manage partial configuration and factory upgrades. This management is included in the PCK and it is transparent for the user.

OSUP Secure Upgrade

You can perform a secure upgrade (fail-safe) of any flash binary section or file included in a Spirit HN software release by creating backup images during the upgrade procedure.

You can perform upgrades using the Layer 3 (FTP, TFTP) or Layer 2 (L2Upgrade).

> **Note:** MaxLinear recommends that you use L2Upgrade only for G.hn nodes connected locally through Ethernet.

The upgrade procedure is secure because it prevents a section from being replaced until a verification that the download and application have been completed without errors. The image to use in the upgrade is called one step upgrade procedure (OSUP) because this upgrade procedure is intended to be performed in one step from the customer's point of view.

The sections available in the OSUP image are:

- Firmware.
- Loader.
- Factory reset files—Set of parameters to restore the configuration during a factory reset. There are three possible modes:
    - Replace_all: All device parameters are restored during a factory reset.
    - Params_update: The device factory reset file is regenerated by changing the values of the parameters included in Params_update and maintaining all other parameter values.
    - Profiles: You can define a subset of parameters to apply regionally during a factory reset.
      You can select the profile to apply with the `FACTORYRESET.PROFILE.ID` parameter.
- Configuration files—Set of parameters associated with the firmware version in the OSUP image. There are three possible modes:
    - Replace_all: All device parameters are replaced with the values configured in the OSUP image. Any previous parameter configuration is lost.
    - Params_update: Only the parameters included in the configuration file are replaced with the values configured in the OSUP image. Any other parameter maintains its value after the upgrade. If a parameter of the old image does not exist in the new firmware version, its value is ported to an equivalent parameter, if it exists thanks to the porting functions implemented in the ConfigLayer of the component to which the parameter belongs.
    - Profiles: You can define a subset of parameters to apply regionally during the upgrade. This is required when different configurations are needed depending on the region, and to maintain other parameters with the value configured by the final customer and when factory reset is not possible.
      You can select the profile to apply with the `FACTORYRESET.PROFILE.ID` parameter.
- User files: You can include in the OSUP image any other file to include in the flash file system as a user file. This feature is useful for customer specific applications such as the web server customization or TR-069 SSL certificates.

Each section of the OSUP image includes a 2-bit mode that indicates when the modem updates the specified file:

- Mode bit 0: Indicates whether the file must be applied immediately or after the next reset.
    - Value 0: Pre-reset, update the files immediately.
    - Value 1: Post-reset, the update applies after the next reset.
- Mode bit 1: Indicates whether a group of files must be processed atomically.
    - Value 0: Individual file.
    - Value 1: The update only applies when all files are downloaded.

Thus, the OSUP image can include one or more of the previous file types, each of them specifying when it must be applied and the sequence.

The OSUP image includes a global header followed by the files, each of them with its own header with extended information. The device downloads them in order and the file is applied according to the configured mode.

You can also include in the OSUP image files already included in the device flash file system to be removed during the upgrade.

OSUP Image Generation

A default OSUP image is provided with the binSDK included in the firmware release, but it is also possible to customize it according to the customers' needs.

The OSUP composer is the tool that generates the OSUP image. Although the OSUP image can be customized manually by editing the configuration files and then calling the OSUP composer, MaxLinear recommends that customers use the graphic tool called product configuration kit (PCK) provided with the Spirit HN release.

The PCK allows customers to perform the most common actions on the OSUP image in a user-friendly manner. It is possible to customize the following elements independently for each product:

- Factory reset values, including options for replace_all, factory_update, and factory profiles.
- Configuration values, including options for replace_all, params_update, and configuration profiles.
- Firmware image encryption.
- Web page interface and product logo image.
- Flash images for mass production.
- Content of the flash image production sector.
- Images for all the different flash sections required to perform an upgrade.

It is also possible to save, recover, and merge different configurations.

After customization, the OSUP images are compiled from the tool itself and are ready to use.

For more information about the PCK, refer to the *G.hn Spirit Firmware Customization Programming Guide* (006PG) or the help embedded in the PCK tool.

Upgrade Process

The upgrade process can be perceived as a single operation, despite the number of files included in the OSUP image, but internally it has two main parts:

- Pre-reset
- Post-reset.

Pre-reset operations perform all actions involved in downloading files or preparing for the required changes. An upgrade operation does not necessarily require downloading a file. It can be, for example, an upgrade of one of the existing configuration files.

The following figure shows the pre-reset actions of the upgrade process.

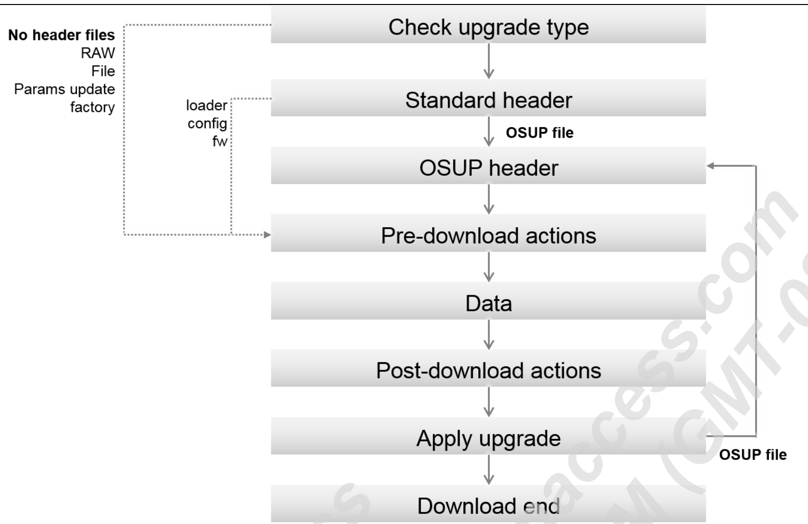


Figure 30: Pre-Reset Actions

Once all the files required to perform the upgrade (or downgrade) are located in the modem, a reset is performed and the post-reset phase takes place.

Throughout the process, a series of auxiliary files are created to keep track of the current state of the process. These files allow you to continue a process that may have been suddenly interrupted (by a power cut, for example) and to prevent file corruption by keeping backup versions of the files while they are modified.

The following figure shows the post-reset actions of the upgrade process.

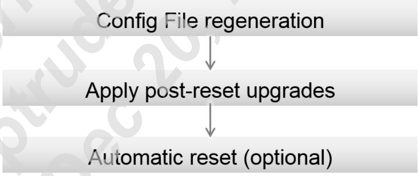


Figure 31: Post-Reset Actions

The expected scenarios are:

- The upgrade failed: All temporary files (including marks, descriptor files, and downloaded files) are removed, so there is no pending action for the next boot.
- The upgrade (during the download) was successful: There may be pending post-reset actions.
- The modem was reset or switched off during the upgrade (during the download): Apply any descriptor file present (represent verified downloads).

Configuration File Regeneration

To ensure the correct functioning of the firmware, make sure that the version of the running firmware and the version of the configuration files are the same.

Otherwise, you must update the configuration file to match the firmware requirements. You must do it when booting the new firmware as follows:

1. Create a temporary configuration file (based on the firmware knowledge).
2. Port data from the outdated configuration file to the temporary configuration file.
   - Upgrade: Take advantage of the deprecated parameters to convert the content to new formats
   - Downgrade: No information on future IDs.
     The previous firmware must create a text file read by any older version to indicate how to complete the necessary parameters.
3. Replace the outdated configuration file.

If you follow this process, an additional reset, performed automatically, is required.

Upgrade Methods

The Spirit HN software supports the following tools to download the OSUP image to the modem:

- Web server: Tool to upgrade the modem locally and remotely.
- TR-069: Tool to upgrade the modem locally and remotely.
- Spirit Configuration Tool (SCT): Tool to upgrade the modem locally.
- DEK (SCP client and L2fwserver): Tool to upgrade the modem locally.

All of these methods allow you to send an OSUP image to the modem.

Note that any progress status displayed by these tools represents the download status, not the upgrade status. Additionally, a reset is required after downloading the OSUP image to complete the upgrade. Then, to check the status of the upgrade, the `FLUPGRADE.GENERAL.STATUS` parameter must be read in the next boot after the upgrade process.

For more information about upgrade methods, refer to the *G.hn Spirit Firmware Customization Programming Guide* (006PG).

Upgrade and Downgrade

Both upgrade and downgrade operations are supported. To ensure compatibility and correct porting of the parameter values, MaxLinear recommends that you upgrade between consecutive versions of the Spirit HN GA, since this operation has been verified. Upgrade between non-consecutive versions of the Spirit HN GA may work but has not been verified.

The Spirit HN software implements a mechanism to keep the modem configuration in upgrade/downgrade operations and to add/remove the new/outdated parameters.

Non-Secure Upgrades

There are other upgrades but they are not secure, that is, it is possible to corrupt the flash if the upgrade is interrupted or the image is corrupted:

- RAW: All flash is erased and replaced with the new flash image.
- Flash: All flash, except the production sector, is erased and replaced with the new flash image. In this case, all production information is maintained.

Factory Reset Parameters

For more information about the factory reset operation, see “Factory Reset” on page 8.

Table 50: Main Factory Reset Configuration Parameters

Parameter	Description
`FACTORYRESET.GENERAL.BUTTON_NUMBER`	Indicates the button assigned to the factory reset functionality.
`FACTORYRESET.GENERAL.BUTTON_TIME`	Indicates the number of seconds you must press the assigned button to perform a factory reset.
`FACTORYRESET.GENERAL.FORCE_RESET`	Forces a factory reset and uses the value passed to set the reset cause.
`FACTORYRESET.GENERAL.PASSWORD`	Password string to allow you to perform a factory reset. It is maintained for backward compatibility, but the final password hash is stored in `FACTORYRESET.GENERAL.PASSWORD_HASH`.
`FACTORYRESET.GENERAL.PASSWORD_HASH`	Password hash to allow you to perform a factory reset.

Support for Configuration Settings Profiles

The Spirit HN software supports different configuration profiles that can be used for country (or region) customizations with factory profiles.

The configuration profile to use during an upgrade process is the same as the one used to apply a factory configuration profile with a factory reset operation (`FACTORYRESET.PROFILE.ID`). It is stored in the MBR (sector 0) and can be set with the production test kit in production time.

You can supersede this configuration with the configuration layer. This enables the configuration of the factory profile by means of the PCK or remotely in runtime by the TR-069 (or LCMP) orders.

To do this, customers must configure the system not to use the profile configured in the MBR and define the profile to use in the configuration file. There is also a configuration parameter to enable the configuration of the profile ID externally (through the LCMP or TR-069).

The following table lists the main configuration layer (CFL) parameters related to factory profiles.

Table 51: Configuration through L2 Parameters for Factory Profiles

Parameter	Description
`FACTORYRESET.PROFILE.SELECT_MBR`	Configures the profile ID source: ■ TRUE: Read-only MBR area, which is configured during manufacturing. ■ FALSE: Read/write user area. **Note:** Remember to unlock this operation (`FACTORYRESET.PROFILE.UNLOCK = YES`)
`FACTORYRESET.PROFILE.UNLOCK`	Locks or unlocks factory profile selection (`FACTORYRESET.PROFILE.ID` and `FACTORYRESET.PROFILE.SELECT_MBR`). This operation is always locked when the modem is switched on.

Table 51: Configuration through L2 Parameters for Factory Profiles (Continued)

Parameter	Description
`FACTORYRESET.PROFILE.ID`	Selects a factory settings profile: ■ n = 0: None. ■ n = 1–255: profile *n'* When writing, set `FACTORYRESET.PROFILE.SELECT_MBR` to *NO*. Otherwise, it returns an error. When reading, the response depends on the profile ID source (`FACTORYRESET.PROFILE.SELECT_MBR`). **Note:** Remember to unlock this operation (`FACTORYRESET.PROFILE.UNLOCK = YES`) **Important:** ■ When the value of the profile ID in flash is modified and `FACTORYRESET.PROFILE.SELECT_MBR = YES`, it is not updated in this parameter after the next reset. ■ If the file for the profile configured in this parameter does not exist in the device, the factory reset process is performed anyway applying only the default factory values. You can detect this situation by checking the value stored in `FACTORYRESET.PROFILE.LAST_CHECK` after the factory reset.
`FACTORYRESET.PROFILE.LAST_CHECK`	Indicates the result of the last factory reset related to the factory profile ID: ■ 0: No factory reset since the last upgrade. ■ 1: Factory profile ID configured and found. ■ 2: Factory profile ID not configured. ■ 3: Factory profile ID configured but not found. ■ 4–255: Reserved.
`FACTORYRESET.GENERAL.COUNTER`	Indicates the number of factory resets performed since a flash or RAW image has been loaded.

When during an upgrade the `FACTORYRESET.PROFILE.ID` parameter has been configured, the upgrade process checks for partial configuration files and only the default one and the one with the same ID as the one configured is applied.

You can generate these partial configuration files with the latest PCK.

After the upgrade, you can know the status of the last upgrade related to the configuration profiles by using the parameters listed in the following table.

Table 52: CFL Parameters Providing Information About Profiles After an Upgrade Process

Parameter	Description
`FLUPGRADE.PROFILE.ID`	Configuration profile selected during the last OSUP upgrade. The value must match the one provided in the `FACTORYRESET.PROFILE.ID` parameter before the upgrade.
`FLUPGRADE.PROFILE.LAST_CHECK`	Indicates the result of the last OSUP upgrade related to the configuration profiles: ■ 0: No OSUP upgrade. ■ 1: Configuration profile ID configured and found. ■ 2: Configuration profile ID not configured. ■ 3: Configuration profile ID configured but not found. ■ 4–255: Reserved.

Firmware Encryption

By default, the firmware image is stored in flash without encryption but you can generate the OSUP image with the firmware encrypted.

The PCK tool allows you to enable or disable the firmware binary encryption in firmware versions that include this feature.

To enable it, select this option in the **Manufacturer ID** section and fill in the password.

After generating the binaries, the firmware to store in flash is encrypted and the loader is configured with an encrypted pass phrase that allows you to decrypt the firmware when booting from flash.

> **Note:** When you use this feature, MaxLinear recommends that you always use OSUP images that include both loader and firmware. If the loader is not updated and it is not configured to decrypt the firmware, the modem is forced to an Ethernet boot recovery.

Pairing Protocol

This feature provides a secure domain setup process that requires minimal intervention and eliminates the need for a computer to configure a secure G.hn network, providing a plug and play experience for the user who uses G.hn nodes from different vendors.

Pairing provides a simple way to establish a secure domain by generating a random domain name (DN) and an encryption key that are exchanged with the nodes that you want to join to the secure domain. Within a secure domain, communications are encrypted.

The process uses a trigger event to enable the creation of a secure G.hn network in a simple way. Such an event generated in one of the nodes within a secure domain informs the domain to open a time window where new nodes can be added to the domain. On the other hand, generating the trigger event in a node outside the domain indicates that a search for a secure domain must be started to access it.

There are several ways to generate this *trigger event*:

- Using a physical push button.
- Through the configuration layer.
- Generating it automatically upon boot (*boot pairing*)

You can configure or preconfigure the DN and the encryption key in different ways. This is initially vendor discretionary. Maxlinear’s G.hn nodes are preconfigured with a default DN and no encryption. In the pairing process, an encryption key is generated randomly if it has not been defined beforehand.

> **Note:** The encryption key is unique for the domain and it is shared by all nodes.

The pairing feature can be configured according to customers’ requirements. The default and recommended pairing mode of operation meets the ITU recommendations and provides the highest level of compatibility with modems from different vendors.

The following states are defined for a node:

- Non-secure active: Communications are not encrypted. All nodes that share the same domain name establish a non-secure domain. The default domain name for this case is *HomeGrid*.
- Unconnected: The domain name is *UNCONNECTED*. The nodes in this state do not try to establish a connection with other nodes. The only way to leave this state is to initiate a pairing operation and establish or join a secure domain with at least another node. If after a pairing operation the node does not find itself in a secure domain with at least another node, it returns to the *unconnected* state.
- Secure node: Communications are encrypted and new nodes can be added to the secure domain by initiating a pairing operation from a node within the domain and a node that wants to enter the domain.

After a successful pairing operation, the nodes store the parameters to join the new domain and the encryption keys. In future boots, they will try to establish or join the same secure domain, until an event to leave the secure domain is received.

In ITU pairing mode, the nodes use the procedures and management messages recommended by the ITU for the pairing operation. You can customize several features through the configuration layer parameters:

- Push buttons and LEDs.
- Boot pairing operation.
- ITU mode.
- One push.
- Pairing end mode.
- Single node.
- Prevent auto-pairing.

The following sections describe these features.

Push Buttons and LEDs

It is possible to configure physical buttons connected to GPIOs to trigger events in order to initiate pairing operations or to leave a secure domain. You can even use a single button and perform different actions depending on the time the button is pressed. You can use the blinking or the color of the LED to signal the operation to perform in each case.

Boot Pairing Operation

When enabled, the nodes are preconfigured in UNCONNECTED mode and they initiate a pairing operation immediately after they are powered on if they are not already part of a secure domain.

ITU Mode

The ITU defines a specific procedure and messages to perform a pairing operation in the *ITU-T G.9978 Standard*. When enabled, the ITU mode is used. This is the default and preferred operating mode for pairing, providing the highest level of interoperability.

One Push

This is a custom pairing flavor, that does not require nodes to leave a secure domain beforehand to initiate a new pairing operation. The first modem where the event is generated defines the target secure domain. The second is generated in the node that needs to move from a different domain to the target secure domain.

Note: This type of pairing applies only to PLC products.

Pairing End Mode

When a pairing operation completes and the node has established a secure domain, but no other node is present in the domain, you can configure it to work in one of the following modes:

- Mode 0: The pairing operation completes with a single node in a secure domain.
- Mode 1: The node goes into the non-secure active state.
- Mode 2: The mode goes into the UNCONNECTED state.

Single Node

When enabled, only one node can join a secure domain during a pairing operation. Then the pairing window closes and to let another node enter the domain, an additional pairing operation must be started. However, there are two exceptions, where several nodes can join the domain in a single pairing window, regardless of the configuration of this parameter:

- In a *boot pairing* operation.
- In a pairing operation performed through *IEEE 1905*.

  Note: Not all combinations of pairing configuration layer parameters are tested. MaxLinear recommends certain of them and customers must perform their own verifications if they want to use a custom set of pairing features.

The following table lists the main configuration layer parameters related to pairing.

Table 53: Pairing Configuration Parameters

Parameter	Description
`PAIRING.GENERAL.BOOT_PAIR_ENABLED`	Indicates whether the node starts an automatic pairing operation at bootup if it is a non-secure node.
`PAIRING.GENERAL.PAIRING_END_MODE`	Indicates what to do if, when the pairing operation finishes, the node is not in a secure domain with at least another node: ■ 0: Generates the secure domain only with one node. ■ 1: Returns to the default domain. ■ 2: Goes to the inactive mode.
`PAIRING.GENERAL.SINGLE_NODE_ENABLED`	Indicates whether the node accepts only one node per pairing operation (except for automatic multinode and 1905 pairing).

To configure the pairing- ITU, the parameters must be set as follows:

```
PAIRING.GENERAL.BOOT_PAIR_ENABLED = YES
PAIRING.GENERAL.PAIRING_END_MODE = 2
PAIRING.GENERAL.SINGLE_NODE_ENABLED = YES
```

Prevent Auto-Pairing

Certain functions have been exported to the API to allow customers to implement prevention of auto-pairing with the runtime firmware. This can be useful during manufacturing to abort the pairing if a particular domain name is detected.

Three new API functions are added:

- `PairingHook`: Hook called on every iteration of the pairing task. You can use it to execute the code as part of the iteration.
- `PairingOngoingAbort`: If a pairing operation is ongoing, it sends an event to abort it.
- `MasterSelectionDetectedDomainsListGet`: It fills a table of domain names with those detected from other neighboring domains.

Pairing External Validator API (ValidatorApp)

The Spirit HN software provides an API to support the use of an external validator entity to accept or reject new nodes that try to enter a secure domain.

A typical use case of such a validator entity is an external application that runs on a mobile phone, but the concept is generic and can apply to any system capable of granting or denying domain access.

A new component, called ValidatorApp, has been included. This component is fully implemented in public code and customers can customize it according to their needs.

This component includes the following elements:

- Basic configuration layer parameters, to enable or disable the functionality and store a MAC address that can correspond to the external validator entity.
- A basic console with a test command.
- A simple task. Currently, it only dumps data every time it is created.
- A small API. It is the main part regarding functionality.

  Note: The ValidatorApp functionality is fully available in the Spirit HN SDK as a source code, which allows customers to extend the functionality and to adapt it to their needs.

For more information about the implementation and how to adapt it, refer to the *G.hn Spirit Firmware Customization Programming Guide* (006PG).

Multicast Support

In Internet Protocol Television (IPTV) networks, the HN network must route multicast video flows based on the IGMP and MLD control traffic coming from the IPTV operator.

IGMP and MLD snooping is used to dynamically configure interfaces so that multicast flows that enter the Spirit HN network are only routed to users who specifically need that traffic flow.

IGMP and MLD Snooping

To efficiently route multicast traffic flows, the Spirit HN software can snoop on both IGMP and MLD protocol packets. This means that:

- Any IGMP v1, v2, or v3 and/or MLD v1, v2 packet is inspected internally, with very little CPU overhead.
- The appropriate routes are created, updated, or deleted according to the packet that is received.

By default, only IGMP snooping is enabled in HN products. However, if necessary, you can enable IGMP and MLD snooping at the same time.

Because IGMP works on IP addresses and the Spirit HN devices are based on MAC addresses, a mapping of the multicast IP address to a destination host MAC is performed.

This IP to MAC conversion is then used to update the bridge with the appropriate route for that MAC address.

The following figure shows the IPv4 to MAC conversion.

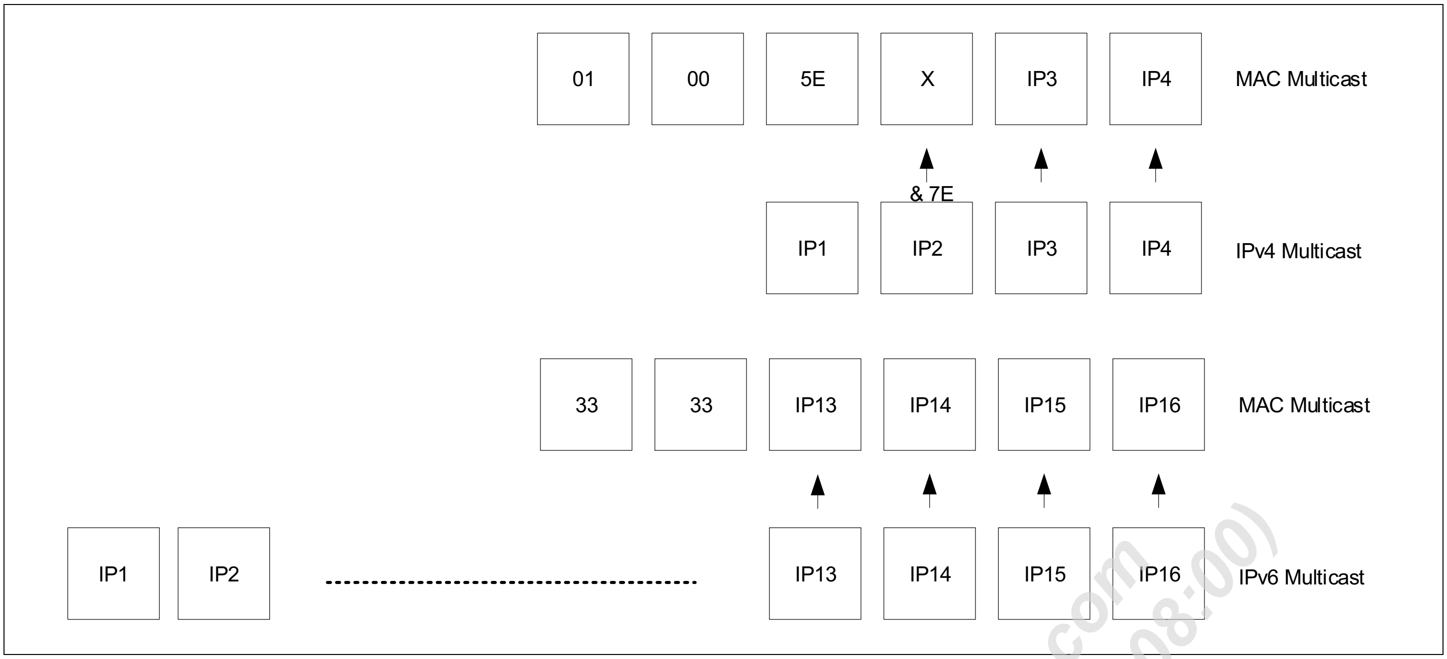


Figure 32: IPv4 Multicast to Multicast MAC Address Conversion

Note: Certain multicast IP addresses share the same MAC address. Client must ensure that the IP addressing scheme establishes a unique MAC address.

Routing Multicast Traffic

Queries are sent between the source host and the receiving host to determine the route that will be taken to identify the video source identity and to enable video streaming.

The video source node is the Spirit HN node that is closest to the video server.

The video source is identified based on the reception of an IGMP, MLD, or a multicast router advertisement packet through an Ethernet port.

If the video source is not identified because of a lack in IGMP, MLD, or multicast router advertisement packets, the Spirit HN network can be configured to:

- Drop reports and leave packets. This is done by default.
- Broadcast reports and leave packets.

Note: A multicast setup must generate queries periodically to determine which groups are still active and able to detect the video source node.

General queries are sent to all Spirit HN nodes. Specific queries are sent only to nodes that have reported an interest in the queried multicast IP.

Reports are used to populate the bridge associating the converted IP multicast address into a MAC with the correct route indication. Reports are sent towards the video source node which is in charge of forwarding the reports towards the video server through its Ethernet interface.

The following figure shows the benefits when IGMP snooping is enabled compared to when it is disabled.

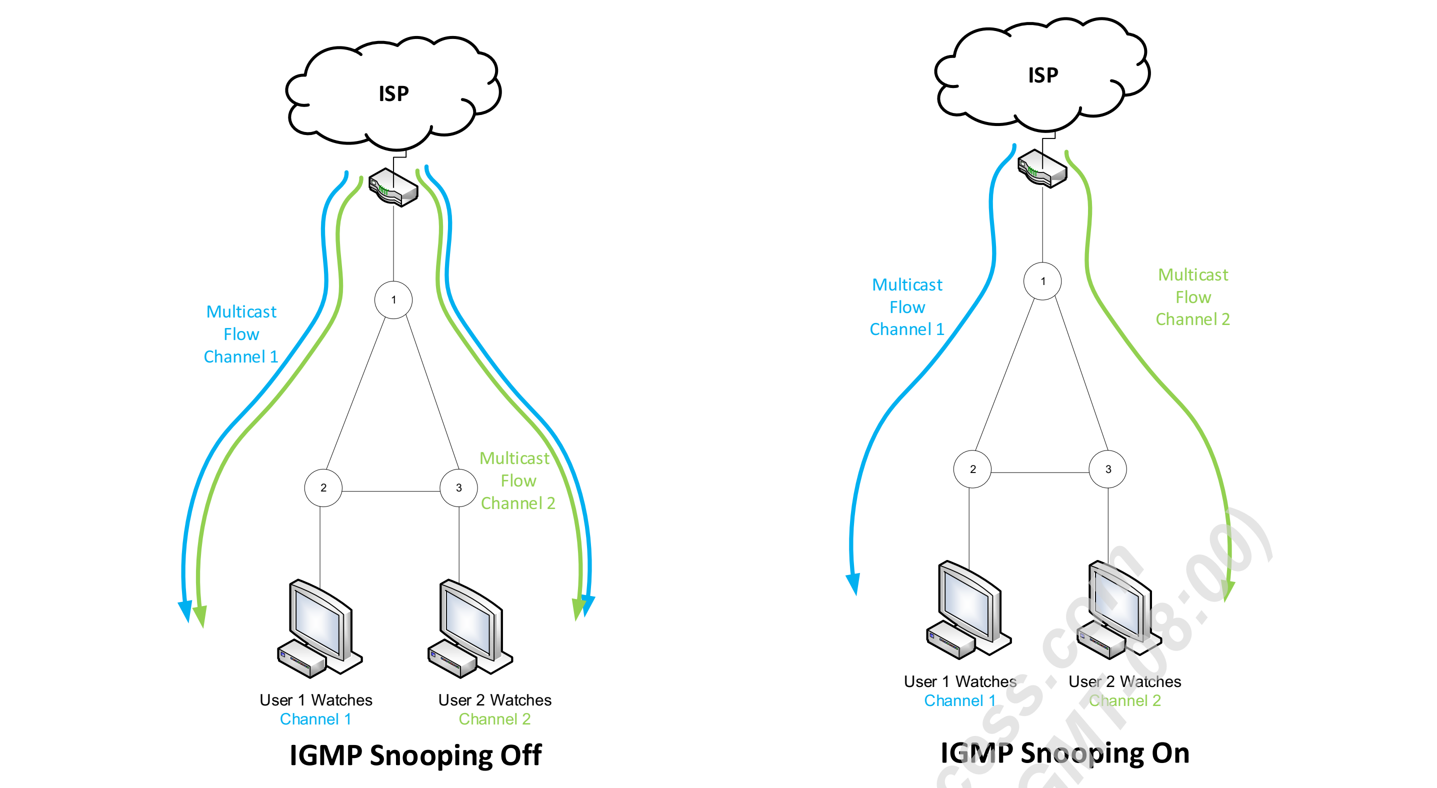


Figure 33: Benefits of IGMP Snooping

Multicast Address Ranges

There are a maximum of four multicast IPv4 address ranges available for configuration. By default, there is only one range defined (224.0.0.0–239.254.255.255).

Any IGMP packet address within the valid ranges is snooped, and its potential associated multicast flow is routed using a unicast address. The IGMP packets and their associated multicast data flows outside the valid range are dropped.

Defining ranges makes it possible to exclude multicast traffic from being routed as multicast traffic. The excluded multicast traffic can be managed through firmware customizations using the SDK API.

Multicast Filtering

This feature consists on dropping traffic that is addressed to a multicast address that is not found, and is neither IGMP nor MLD, leaving the bandwidth to be used only by solicited videos.

Multicast traffic suppression is enabled by default in the HN. Unsubscribed multicast addresses not included in exceptions are dropped to prevent network flooding.

Table 54: Unknown Multicast Traffic Suppression Parameter

Parameter	Description
`MCAST.GENERAL.MCAST_FILTERING_ENABLE`	Enables the multicast filtering feature. If enabled, all unsolicited multicast traffic is blocked. In IPv4 multicast traffic, only traffic between the IP ranges defined in the `MCAST.GENERAL.IGMP_IP_RANGES_DEF` parameter is blocked if unsolicited. **Important:** This feature involves a higher CPU load, so MaxLinear recommends that you enable it only in the video source. Only 100Kbps of broadcast traffic can be managed in this mode.

IGMP and MLD Fast Leave

This feature consists on an immediate blocking of a multicast group for a particular port when a *LEAVE* message is received from that port.

This feature is disabled by default in the Spirit HN software and it is not validated.

The following table lists the corresponding configuration parameter.

Table 55: Multicast Fast Leave Configuration Parameter

Parameter	Description
`MCAST.GENERAL.FAST_LEAVE_ENABLE`	If the value is: ■ YES: When a *Leave Group* message is received from a specific port (G.hn or Ethernet), the multicast stream forwarding for this port is immediately blocked. ■ NO: When a *Leave Group* message is received, the multicast stream is forwarded until three group-specific or general queries are sent for the group and no reports are received.

Multicast Exceptions

In the multicast component, you can define a range of IP addresses to be sniffed and managed by the IGMP component. Certain protocols such as mDNS use a multicast IP address within this range and they are not handled as expected.

To manage this situation, a new procedure to define up to 20 ranges of IP exceptions (10 for IGMP and 10 for MLD) has been added using the following parameters:

- `MCAST.GENERAL.IGMP_IP_EXCEPTION`
- `MCAST.GENERAL.MLD_IP_EXCEPTION`

Each multicast IP exception is created as an `ENABLE` + `EXCEPTION_IP` + `EXCEPTION_MASK` group.

When there is an incoming multicast packet, whose `INCOMING_IP` is inside the `MCAST.GENERAL.IGMP_IP_RANGES_DEF` or `MCAST.GENERAL.MLD_IP_RANGES_DEF`, it is matched (AND operation) against the `EXCEPTION_MASK` of each of the defined exceptions in `MCAST.GENERAL.IGMP_IP_EXCEPTION` (or `MCAST.GENERAL.MLD_IP_EXCEPTION`). If the `INCOMING_IP <AND> EXCEPTION_MASK AND` operation equals the `EXCEPTION_IP` of the exception, the incoming packet is treated as a non-IGMP (or non-MLD) packet. This behavior enables the creation of exceptions for individual IP addresses and IP groups.

Notes:

- `EXCEPTION_MASK` and `EXCEPTION_IP` pairs must be consistent. If the `IP` and `MASK` are chosen so that the `EXCEPTION_MASK <AND> INCOMING_IP` operation never matches `EXCEPTION_IP`, the rule is never applied.
- If the defined exception is outside the ranges defined in `MCAST.GENERAL.IGMP_IP_RANGES_DEF` and `MCAST.GENERAL.MLD_IP_RANGES`, the exception is ignored.

Configuration

A multicast IP exception is defined as a configuration layer (CFL) parameter. The CFL parameter has a length of 9 bytes, which are divided into three groups:

- Byte 1 → `ENABLE` → 0 means the exception is disabled, 1 means the exception is applied.
- Bytes 2–5 → `EXCEPTION_IP` → 4 bytes that define the result between `INCOMING_IP <AND> EXCEPTION_MASK`. It is like the result of applying the rule.
- Bytes 6–9 → `EXCEPTION_MASK` → 4 bytes that define the mask to add to `INCOMING_IP`. It is like the rule.

IGMP Example

- `MCAST.GENERAL.IGMP_IP_EXCEPTIONS.1.0 = 0, x,x,x,x, x,x,x,x` to disable this `IP`/`MASK`.
- `MCAST.GENERAL.IGMP_IP_EXCEPTIONS.5.0 = 1, 224,5,230,0, 255,255,255,0` considers as exceptions all 224.5.230.x
- `MCAST.GENERAL.IGMP_IP_EXCEPTIONS.9.0 = 1, 224,5,0,0, 255,255,0,0` considers as exceptions all 224.5.x.x
- `MCAST.GENERAL.IGMP_IP_EXCEPTIONS.7.0 = 1, 224,7,230,211, 255,255,255,255` considers as exception 224.7.230.211
- `MCAST.GENERAL.IGMP_IP_EXCEPTIONS.6.0 = 1, 224,7,230,200, 255,255,255,248` considers as exception from 224.7.230.200 to 224.7.230.207
- `MCAST.GENERAL.IGMP_IP_EXCEPTIONS.10.0 = 1, 224,5,230,0, 100,255,255,0` is a non-consistent configuration.

  Note: The same settings exist for `MCAST` IPv6 packets. `MCAST.GENERAL.MLD_IP_EXCEPTIONS` takes into account that IPv6 addresses have 16 bytes instead of 4 bytes.

Multicast Summary

The Spirit HN software supports the following:

- IGMPv1 (RFC1112).
- IGMPv2 (RFC2236).
- IGMPv3 (RFC3376 + RFC5790). For more information, see the note below.
- MLDv1 (RFC2710).
- MLDv2 (RFC3810 + RFC5790). For more information, see the note below.
- Multicast Router Solicitation.
- Multicast Router Advertisement.
- IGMP and MLD fast leave.
- Multicast video source mode.
- Four ranges of addresses for multicast operation.
- The maximum number of multicast channels supported is 128.

  Note:

  - The current implementation of IGMPv3 and MLDv2 is based on recommendations described in the *RFC 5790 Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2)* protocols. All report packages that display `IS_IN {x}`, `TO_IN {x}`, `ALLOW{x}`, `IS_EX { }` are considered *REPORT* messages. Other message are considered *LEAVE* messages.
  - Source filtering is not supported.

Table 56: Multicast Configuration Parameters

Parameter	Description
`MCAST.GENERAL.FAST_LEAVE_ENABLE`	If the value is: ■ YES: When a *Leave Group* message is received from a specific port (G.hn or Ethernet), the multicast stream forwarding for this port is blocked. ■ NO: When a *Leave Group* message is received, the multicast stream is forwarded until three group-specific or general queries are sent for the group and no reports are received
`MCAST.GENERAL.FORCED_DM_ENABLE`	■ If enabled, the node connected to the gateway is configured as the video source and queries from other nodes are ignored. ■ If disabled, the video source is set according to the queries received.
`MCAST.GENERAL.FORCED_GW_ENABLE`	■ If enabled, the node sends ICMP router discovery packets while the video source has not yet been detected. ■ If disabled, the node does not send any ICMP router discovery packet.
`MCAST.GENERAL.IGMP_ENABLE`	Set to enable or disable the IGMP snooping.
`MCAST.GENERAL.MCAST_FILTERING_ENABLE`	Enables the multicast filtering feature. If enabled, all unsolicited multicast traffic is blocked. In IPv4 multicast traffic, only traffic between the IP ranges defined in the `MCAST.GENERAL.IGMP_IP_RANGES_DEF` parameter is blocked if unsolicited. **Important:** This feature involves a higher CPU load, so MaxLinear recommends that you enable it only in the video source. Only 100 Kbps of broadcast traffic can be managed in this mode.
`MCAST.GENERAL.MCAST_FILTERING_IN_VS`	If the value is: ■ YES: If the multicast filtering feature is enabled, it is only activated in the video source. ■ NO: If the multicast filtering feature is enabled, it is activated in all nodes.
`MCAST.GENERAL.MLD_ENABLE`	Set to enable or disable the MLD snooping.
`MCAST.GENERAL.REPORT_BROADCAST_ALLOWED`	Report broadcast behavior. ■ YES: The node is configured to broadcast reports depending on the configuration in the `MCAST.GENERAL.REPORT_BROADCAST_MODE` parameter. ■ NO: Reports are sent only to the video source if it is known. The term *video source* refers to the node whose Ethernet port is connected directly to the home gateway. MaxLinear recommends that you set it to *NO* in IGMP v1 and v2 scenarios with more than one set-top box (STB). Otherwise, the broadcasted reports can prevent other STBs from joining the reported channel because they think there is no need to transmit the report packet again.
`MCAST.GENERAL.REPORT_BROADCAST_MODE`	Report broadcast forwarding behavior when the `MCAST.GENERAL.REPORT_BROADCAST_ALLOWED` parameter is enabled. ■ 0: Broadcast reports only when the video source is unknown. ■ 1: Broadcast reports always. ■ 2: Broadcast reports always if IGMPv3 and only when the video source is unknown in others. The term *video source* refers to the node whose Ethernet port is connected directly to the home gateway.
`MCAST.GENERAL.VIDEO_SOURCE_MODE[10]`	Video source mode: ■ AUTO: A query from the PLC or external interface transforms the node into a video source. ■ FORCED: A query from the PLC is ignored and a query from the external interface transforms the node into a video source. ■ FORBIDDEN: A query from the external interface is ignored. The node will never be a video source.

Standby Modes

This feature saves energy when the modem does not transmit data and enable designs based on Maxlinear's digital baseband processor and analog front-end to be compliant with the *European Directive 2005/32/CE* (Ecodesign), also known as *ErP*.

When a G.hn node does not show Ethernet activity and does not relay traffic between other nodes, it enters a low power consumption state. In this state, the node neither transmits nor receives frames, so it is virtually switched off for the rest of the network. This functionality can work with the short-term and long-term power saving modes defined in G.hn.

You can configure the Ethernet conditions with the configuration layer parameters of the `POWERSAVING` group.

There are three different modes:

- Standby only: The modem enters in standby only by pressing the **Standby** button. Maxlinear's reference designs do not include this button, but the firmware does support it.
- Ethernet link: The modem enters in standby when the Ethernet link is down and the medium conditions are also met.
- Ethernet activity: The modem enters in standby when the medium conditions are met and there is no traffic flowing through Ethernet, even with link up. In this case, the modem only wakes up when the Ethernet traffic flow resumes—not if the traffic goes through the G.hn link.

The medium conditions for the node are:

- Not to be a domain master for other nodes.
- Not to be a relay node for data or *Medium Access Plan* (MAP) messages.

Both conditions, medium and Ethernet, must be maintained for an `IDLE_TIME` that can be configured in the `POWERSAVING` parameters. By default, its value is 300 seconds. The standby enter/exit triggered by the button is immediate. The coverage LED blinks once every five seconds when the power saving mode is active. You can also configure this behavior with the `POWERSAVING` parameters.

Table 57: Powersaving Configuration Parameters

Parameter	Description
`POWERSAVING.BUTTON.HOLD_TIME`	Specifies how long the button must be pressed (before being released) to enter/exit the power saving state.
`POWERSAVING.BUTTON.NUMBER`	Button number used to enter and exit the *deep* power saving mode. It is the index in the `UI.BUTTON.GPIO_NUMBER` where the button is defined.
`POWERSAVING.GENERAL.IDLE_TIME`	Time that the node must be idle before entering the power saving mode.The idle time is computed when the condition specified in the `POWERSAVING.GENERAL.MODE` parameter is met. A node may not automatically enter the power saving mode when it is a domain master with more than one end point.
`POWERSAVING.GENERAL.MODE`	Criterion used to consider the node in idle state: ■ 0: Disabled. ■ 1: Ethernet link. The idle time is computed from the moment the Ethernet link goes down. Reactivation takes place when the link is up or the button is pressed. ■ 2: Ethernet activity. The idle time is computed from when the last Ethernet packet is transmitted or received. Reactivation takes place when a packet is received or the button is pressed.

Table 57: Powersaving Configuration Parameters (Continued)

Parameter	Description
`POWERSAVING.GENERAL.STATUS`	Current operational state of the node: ■ 0: Full-power mode (L0). It can transmit and receive network traffic at its maximum capacity. ■ 1: Efficient-power mode (L1). It can transmit and receive network traffic at its maximum capacity. ■ 2: Low-power mode (L2). It can transmit and receive network traffic at reduced capacity. ■ 3: Idle mode (L3). It cannot transmit and receive network traffic, but can transmit and receive management traffic. ■ 4: The node is down. ■ 5: The node is turned on, but not ready to transmit and receive network traffic. ■ 6: The node is in a fault/error condition. ■ Other: Reserved by ITU-T. For more information, refer to the *ITU-T G.9962 Standard*.
`POWERSAVING.GPIO.VALUE_DURING_POWERSAVING`	Configuration value in the GPIO register during power saving/standby state. Every bit of the register corresponds to a GPIO number, where the least significant bit (LSB) corresponds to GPIO 0 and the most significant bit (MSB) corresponds to GPIO 31. This is the configuration value that sets all the LEDs and other external devices to the OFF state during power saving to reduce power consumption. The only GPIO that is not affected by this parameter value is the one indicated by the `POWERSAVING.GPIO.LED_BLINK` parameter. The LED controlled by this GPIO can be switched between ON and OFF states during power saving state. When the modem detects the `EXIT` condition, the GPIO indicated by the `POWERSAVING.GPIO.LED_EXIT` parameter (if configured) is set and the rest of the GPIOs retain the value indicated by the `POWERSAVING.GPIO.VALUE_DURING_POWERSAVING` parameter until the modem exits the power saving state. For the GPIOs that control LEDs, the behavior of the LED depends on the reference design implementation. For the MaxLinear reference design, the default value of this parameter (`0xFFFFFFFF`) switches off all LEDs.
`POWERSAVING.LED.BLINK_NUMBER`	Specifies the number of LEDs to which the standby LED is connected. This change takes effect immediately. It is the index in the `UI.LED.GPIO_NUMBER` where the button is defined.
`POWERSAVING.LED.BLINK_OFF`	Specifies the amount of time in milliseconds that the standby LED is off. This parameter is combined with `POWERSAVING.LED.BLINK_ON` to define the blinking pattern during standby mode.
`POWERSAVING.LED.BLINK_ON`	Specifies the amount of time in milliseconds that the standby LED is on. If the `POWERSAVING.LED.BLINK_OFF` parameter has a value other than 0, the standby LED blinks with the specified time ratio. Otherwise, it remains solid.
`POWERSAVING.LED.EXIT_NUMBER`	Specifies the number of LEDs to which the LED that indicates the exit of the power saving state is connected. This LED indicates the return to normal operation during the time it takes for the firmware to recover from the suspended state. This change takes effect immediately. It is the index in the `UI.LED.GPIO_NUMBER` where the button is defined.

This feature can be confusing if the modem is not connected through the Ethernet port, since the modem *disappears* from the channel. For testing, you can temporally disable this feature through the SCP or Web.

The modem enters in low power consumption mode either manually, by means of the push button, or automatically, by detecting the status of the Ethernet link or activity. You can find the power consumption measurements of all G.hn EVK designs in different operation modes in the hardware test reports for PLC, Coax, and Phone in the myMxl Secure Site (www.maxlinear.com).

Note: You cannot remotely wake up a node in standby mode.

IEEE 1905.1a

The Spirit HN supports the *IEEE 1905.1a* protocol as a high-level layer to simplify the management of hybrid networks that can mix different technologies such as WiFi, MoCA, or G.hn. It allows you to gather information about all devices in the network, topology, and resources that can improve network setup, configuration, and maintenance.

The IEEE 1905 group of the L2 parameters includes the parameters to enable and configure the IEEE 1905.1a.

The following figure shows the IEEE 1905 abstraction layer.

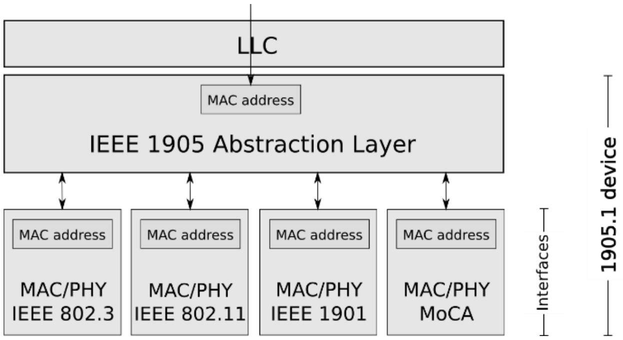


Figure 34: IEEE 1905.1 Abstraction Layer

As defined in the *IEEE 1905.1a Standard*, the IEEE 1905 entity included in G.hn modems requires a specific MAC address to address the *IEEE 1905* messages. It cannot be its own firmware MAC address which is already configured during production and can be checked later in the `SYSTEM.PRODUCTION.MAC_ADDR` parameter.

You can configure the second MAC address that the IEEE 1905.1a uses during production in the same way as its own firmware MAC address and check it in the `SYSTEM.PRODUCTION.MAC_ADDR2` parameter.

When you include the IEEE 1905.1a in already deployed devices, you cannot configure this new second MAC address in the production sector. In this case, the second MAC address is formed by modifying 3 MSB bytes of its own fimware MAC address according to the configuration stored in `IEEE1905.GENERAL.AL_MAC_MSB_DEFAULT`.

Some of the main benefits from the IEEE 1905 1a are:

- Fallback: When a link temporarily goes down or is congested, an alternative route is used, if there are multiple routes between the end points.
- Aggregated throughput: Use multiple interfaces to maximize throughput.
- Multiple simultaneous streams: With applications such as interactive TV, one person can watch multiple streams simultaneously.
- Load balancing: Intelligently distribute multiple video streams over different paths to limit congestion and maintain reliability.
- Quality of service (QoS): Uniformed prioritized QoS on multiple technologies.
- Managing network traffic under fluctuating interference: Enables a heterogeneous network to operate in a way that allows rerouting of established flows when the links on which they are do not support the QoS (for example) requirements of the flow. It is very important for PLC and—mainly—Wi-Fi.
- Security: P1905.1 allows you to configure the devices in the same way, for example with a simple button push.

Table 58: IEEE 1905 Configuration Parameters

Parameter	Description
`IEEE1905.GENERAL.ALME_TCP_PORT`	TCP port number where the abstraction layer entity listens for *ALME* messages. It only takes effect after a reset.
`IEEE1905.GENERAL.AL_MAC`	Read-only parameter that contains the AL MAC address currently used by the IEEE 1905 abstraction layer entity. To *set* the AL MAC address, use the `IEEE1905.GENERAL.AL_MAC_MSB_DEFAULT` parameter.
`IEEE1905.GENERAL.AL_MAC_MSB_DEFAULT[3]`	You can use this parameter to *set* the AL MAC address, which you can later *check* by reading the `IEEE1905.GENERAL.AL_MAC` parameter. If the `SYSTEM.PRODUCTION.MAC_ADDR2` parameter contains a non-empty (all zeros or all `0xFs`) address, the AL MAC is set to this value, and the `IEEE1905.GENERAL.AL_MAC_MSB_DEFAULT` parameter is ignored. Otherwise, the AL MAC is set to the three bytes contained in the `IEEE1905.GENERAL.AL_MAC_MSB_DEFAULT` parameter followed by the last three bytes of the *standard* modem MAC address (the `SYSTEM.PRODUCTION.MAC_ADDR` parameter). **Note:** If the `IEEE1905.GENERAL.AL_MAC_MSB_DEFAULT` parameter contains three zero bytes, the IEEE 1905 component does not start unless the `SYSTEM.PRODUCTION.MAC_ADDR2` parameter contains a valid address.
`IEEE1905.GENERAL.ENABLE`	If set to *NO*, the IEEE 1905 component does not start on the next boot. This parameter can be read to check if this component has been properly started.
`IEEE1905.GENERAL.MAP_WHOLE_NETWORK`	■ YES: The abstraction layer entity queries all IEEE 1905 nodes in the network. It requires a large amount of memory. ■ NO: Only immediate neighbors are queried. It only takes effect after a reset.

TR-069

Introduction

This section describes the evaluation TR-069 clients included into the Spirit HN software package. One of the TR-069 clients delivered as an integration example is from *Works Systems, Inc.* and the other is from *Gatespace Networks, Inc*.

> **Note:** Both TR-069 agents included in the Spirit HN software are third-party software provided either by *Works System, Inc.* or by *Gatespace Networks, Inc.* and released by MaxLinear for evaluation purposes only.

Other TR-069 agents can be integrated by customers using the Spirit source SDK.

This section also describes how to configure the MaxLinear G.hn modems to access the Auto Configuration Server (ACS).

The following figure shows the Internet Protocol Television (IPTV) service with the TR-069 platform and the PLC home network.

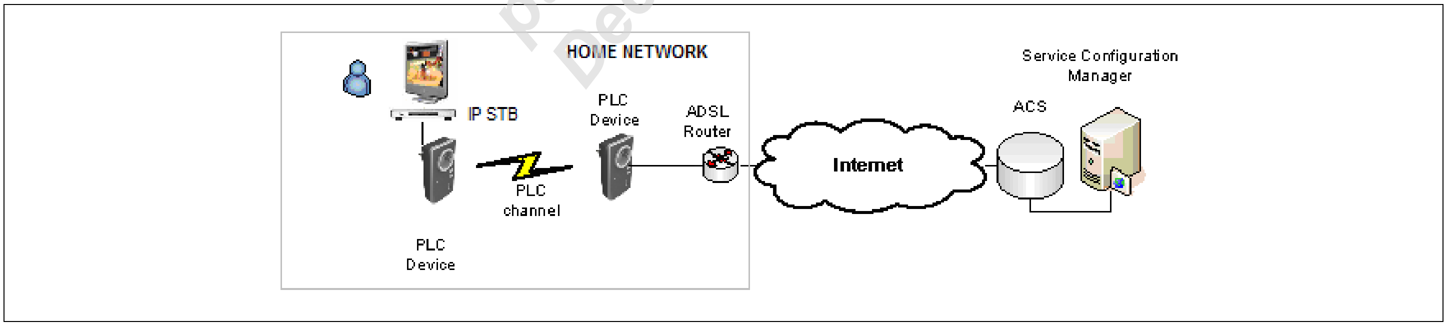


Figure 35: IPTV Service with TR-069 Platform and PLC Home Network

The following technical specifications from the *Broadband Forum's TR-069 standard* have been implemented into G.hn devices to allow remote management:

- TR-069 Amendment 3 = *CPE WAN Management protocol*. TR-069 is a protocol for communication between a customer premise equipment (CPE) and an Auto Configuration Server (ACS) that encompasses secure auto configuration as well as other CPE management functions within a common framework.
- TR-111 = Applying TR-069 to *Remote Management of Home Networking Devices, Part 1 and Part 2 (STUN)*.
- TR-181 Issue 2 Amendment 6 = Device data model for TR-069 for G.hn devices as specified in the *ITU-T G.9962 Unified high-speed wire-line based home networking transceivers — Management specification*. It also includes several vendor-specific objects for MaxLinear's G.hn devices.

MaxLinear's implementation implies that a native TR-069 client can be integrated into G.hn devices without the need to change the home gateway firmware to manage the home network remotely.

Architecture

Protocol Components

The CPE Management Protocol comprises unique components in addition to standard protocols. The following table lists the protocol stack defined by the CPE Management Protocol.

Table 59: TR-069 Protocol Stack

CPE/ACS Management Application
RPC Methods
SOAP
HTTP
SSL
TCP/IP

A standard XML-based syntax is used to encode remote procedure calls.

Secure Socket Layer 3.0 (SSL 3.0) is the standard Internet transport layer security protocol.

In addition to the standard TCP/IP stack, the following specific protocols are used:

- Domain Name Server (DNS) = This functionality is supported in the firmware but you must configure it if the Dynamic Host Configuration Protocol (DHCP) is not used.
- Network Time Protocol (NTP) = This functionality is supported in the firmware but you must configure it separately.

Security Mechanisms

This component allows a high level of security in all message exchanges between the CPE and the ACS. The following security mechanisms are implemented:

- SSL for communication transport between CPE and ACS. This provides transaction confidentiality and data integrity.
- The HTTP layer provides an alternative method for CPE authentication based on shared secrets.

The use of these methods is optional.

TR-069 Client

Default Images

The Spirit HN package provides evaluation products that include the TR-069 functionality by default. The names of the TR-069 evaluation products are as follows:

- <reference design>_WorkssysEval_v1_x
- <reference design>_GatespaceEval_v1_x

Agent Configuration Using PCK

Evaluation images included in the Spirit HN package contain default MaxLinear's TR-069 and basic configuration settings. To generate customized flash images with the appropriate settings, use the PCK and set TR-069 and other basic configuration settings based on their requirements.

Main Configuration Parameters

There are several parameters that can be configured but, depending on the ACS, not all of them are necessary.

The TR-069 parameters can be easily configured using the PCK.

For more information, refer to the *G.hn Spirit Firmware Customization Programming Guide* (006PG).

MaxLinear Vendor-Specific Parameters

MaxLinear vendor-specific parameters have a structure of `X_MAXLINEAR_COM_<parameter name>` and each TR-069 parameter corresponds to a configuration layer parameter. For more information about the configuration layer parameters, refer to **<product>_WorkssysEval_v1_x_parameters_description.html** or **<product>_GatespaceEval_v1_x_parameters_description.html** included in the PCK.

For more information, refer to the *G.hn Spirit Firmware Customization Programming Guide* (006PG).

RPC Methods

The following table lists the main RPC methods implemented in the MaxLinear platform.

Table 60: **RPC Methods**

RPC Methods
Download file type 1 (OSUP file)
Download file type 3 (SCP configuration file)
Upload file type 1 (vendor configuration file)
Upload file type 2 (vendor log file)
Factory reset
Reboot
`GetRPCMethods`, `GetParameterNames`, `GetParameterValues`, and `GetParameterAttributes` (retrieving data from the client)
`SetParameterValues` and `SetParameterAttributes` (setting data to the client)

TR-111 Part 1

The TR-069 client and the DHCPv4 client support also the *TR-111 part 1 (Device-Gateway Association)*:

- The modem includes the DHCP option 125 including the device identity information for each `DHCPDISCOVER`, `DHCPREQUEST`, and `DHCPINFORM`.
- The gateway must also include the DHCP option 125 including the gateway identity information for each `DHCPOFFER` and `DHCPACK` so that the modem can extract this information and copy it to the corresponding parameters.
- The modem sends active notifications if any of the parameters in `Device.GatewayInfo` are modified.
- The modem also sends `DHCPINFORM` messages whenever the TR-069 client is included in the firmware and the DHCP is not activated. It sends these `DHCPINFORM` messages as long as the configured gateway does not answer. If the modem receives a `DHCPACK` from a gateway other than the one configured, the answer is ignored (the IP address of the configured gateway can be read from `Device.Routing.Router.1.IPv4Forwarding.1.GatewayIPAddress`).

TR-111 Part 2 (STUN)

The integration of the Session Traversal Utilities for NAT (STUN) client into the TR-069 agent enables the use of the *TR-111 part 2* in MaxLinear's G.hn adapters. This functionality allows the operator to initiate the connection from the ACS/STUN server to the PLC adapter at any time using the CPE notification mechanism.

You can use the `TR069.MANAGEMENTSERVER.STUN_<name>` configuration layer parameters to configure the STUN settings. The STUN settings must be configured depending on the operator network scenario using the PCK to generate the appropriate flash image for deployment. The following table lists the default contents.

Table 61: STUN Parameters in TR-069 Data Model

Name	Type	Write	Description
`Device.ManagementServer`	Object	-	This object contains parameters relating to the association of the CPE with an ACS.
`UDPConnectionRequestAddress`	String(256)	-	Address and port to which an ACS may send a UDP connection request to the CPE. It must be in `host:port` or `host`. ■ If `STUNEnable` is true, the *host* and *port* portions of this parameter must represent the public address and port corresponding to the NAT binding through which the ACS can send UDP *Connection Request* messages (once this information is learned by the CPE through the use of STUN). ■ If `STUNEnable` is false, the *host* and *port* portions of the URL must represent the local IP address and port on which the CPE listens for UDP *Connection Request* messages.
`STUNEnable`	Boolean	W	Enables or disables the use of STUN by the CPE. This only applies to the use of STUN in association with the ACS to allow UDP *Connection Request* messages.
`STUNServerAddress`	String	W	Host name or IP address of the STUN server for the CPE to send binding requests if STUN is enabled via `STUNEnable`. ■ If empty and `STUNEnable` is true, the CPE must use the address of the ACS extracted from the host portion of the ACS URL.
`STUNServerPort`	UnsignedInt	W	Port number of the STUN server for the CPE to send binding requests if STUN is enabled via `STUNEnable`. By default, this should equal the default STUN port, 3478.
`STUNUsername`	String(256)	W	■ If non-empty, the value of the STUN USERNAME attribute to use in binding requests (only if the message integrity has been requested by the STUN server). ■ If empty, the CPE must not send STUN binding requests with the message integrity.
`STUNPassword`	String(256)	W	The value of the STUN password to use in computing the MESSAGE-INTEGRITY attribute to use in binding requests (only if the message integrity has been requested by the STUN server). When read, this parameter returns an empty string, regardless of the actual value.
`STUNMaximumKeepAlivePeriod`	Int[-1:]	W	If STUN is enabled, the maximum period, in seconds, that the CPE must send STUN binding requests to maintain the binding in the gateway. This specifically applies to binding requests sent from the UDP Connection Request address and port. A value of –1 indicates that no maximum period is specified.

Table 61: STUN Parameters in TR-069 Data Model (Continued)

Name	Type	Write	Description
`STUNMinimumKeepAlivePeriod`	UnsignedInt	W	If STUN is enabled, the minimum period, in seconds, that the CPE may send STUN binding requests to maintain the binding in the gateway. This limit applies only to binding requests sent from the UDP Connection Request address and port, and only those that do not contain the BINDING-CHANGE attribute.
`NATDetected`	Boolean	-	If STUN is enabled, this parameter indicates whether or not the CPE has detected the address and/or port mapping in use. A true value indicates that the MAPPEDADDRESS received in the most recent binding response differs from the source address and port of the CPE. If `STUNEnable` is false, this value must be false.

Legal Notices

This section provides information about third-party software content related to TR-069 and its legal rights.

Embedded TR-069 Agent from Works Systems, Inc.

This software is provided as an evaluation code and executables. The use of the TR-069 agent in production is not allowed and requires a TR-069 agent license from *Works Systems, Inc*.

> **Note:** This license agreement must be agreed directly with *Works Systems, Inc.*

Embedded TR-069 Agent from Gatespace Networks, Inc.

This software is provided as an evaluation code and executables. The use of the TR-069 agent in production is not allowed and requires a TR-069 agent license from *Gatespace Networks, Inc*.

> **Note:** This license agreement must be agreed directly with *Gatespace Networks, Inc.*

mbedTLS SSH and TLS Stack

The default SSL stack included in TR-069 products is an mbedTLS stack designed for embedded devices. The integrated version 2.16.0 supports SSL version 3 and TLS versions 1.0, 1.1, and 1.2 and other cryptographic capabilities.

It is open source and since it is in public code in the API, customers can customize it if needed. For more information, go to the mbedTLS website (https://tls.mbed.org/) and https://github.com/ARMmbed/mbedtls.

MAC Discover using LCMP Frames

The process to discover the MAC of the G.hn device connected to a host through Ethernet is called MAC discover. It uses SNAP frames sent to the control MAC defined by the ITU for G.hn devices 0x01:0x19:0xA7:0x52:0x76:0x96 and the G.hn responds to them, which allows the host to obtain the G.hn MAC address.

Certain scenarios or network devices cannot correctly handle the SNAP format of the discovery packets. Therefore, a new tool has been developed to perform the discovery process using the standard LCMP packet format and parameters from the HomeGrid forum data model.

Most development tools provided with the Spirit HN software, such as the configuration layer command line tool, the Spirit Configuration Tool (SCT), or the production test kit (PTK) still use SNAP-based discovery. Only the G.hn flashless services include LCMP-based discovery.

LCMP-based discovery is included in a command line tool named `HgfTool` and it is provided in source code.

Command to discover the local G.hn node:

```
$ sudo ./HgfTool -o DISCOVER -i enx00249b135b42
Device discovered: 00:13:9d:00:05:bb
```

Command to discover all G.hn nodes in the network:

```
$ sudo ./HgfTool -o DISCOVER_ALL -i enx00249b135b42
Number of devices found: 1
      Entry: 0    MAC address: 00:13:9d:00:05:bb
```

A more detailed HgfTool usage is explained in its –help option, with the following output:

```
$ ./HgfTtool --help
USAGE:
   ./HgfTool  -o <DISCOVER|DISCOVER_ALL|FACTORY_RESET|READ|RESET> -i
              <interface> [-d <MAC address>] [-p <name>] ...  [-t
              <timeout>] [--] [--version] [-h]
Where:
   -o <DISCOVER|DISCOVER_ALL|FACTORY_RESET|READ|RESET>,  --operation
      <DISCOVER|DISCOVER_ALL|FACTORY_RESET|READ|RESET>
     (required)  Operation to perform.
   -i <interface>,  --interface <interface>
     (required)  Name of the network interface to use.
   -d <MAC address>,  --device <MAC address>
     Device's MAC address.
   -p <name>,  --parameter <name>  (accepted multiple times)
     Name of the HomeGrid Forum parameter.
   -t <timeout>,  --timeout <timeout>
     Response timeout in milliseconds. Default value: 1000 ms
   --,  --ignore_rest
     Ignores the rest of the labeled arguments following this flag.
   --version
     Displays version information and exits.
   -h,  --help
     Displays usage information and exits.
```

Keep-Alive Monitoring

In certain markets, a device hang is critical, for example:

- There is no reset button.
- The device cannot be power cycled.
- Accessing the modem is difficult.

Although the existing watchdog component can cover a large number of cases, keep-alive goes one step further because:

- It makes sure that the device is capable of transmitting and receiving data.
  For its part, the watchdog component makes sure that the firmware still runs in the device, which is different.
- Since it is connected to an external watchdog (not mandatory), it can power cycle the whole board—not only the digital Asic.

The keep-alive monitoring has two main parts:

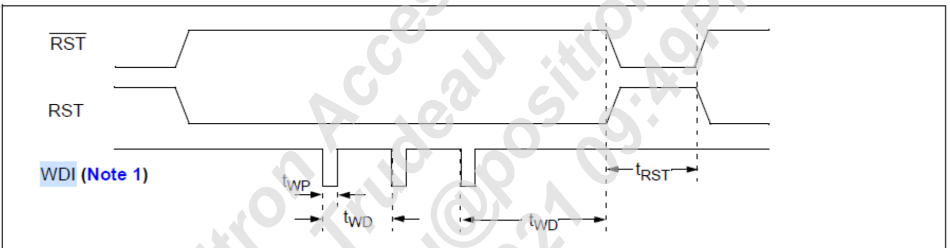
- Enable and feed an external watchdog (if available in the board).
- Monitor certain selected parameters to decide if the device still have connectivity within the network.

  Note: The keep-alive component (mainly decisions about network stability) was not designed for the HN market, so it requires customization of these rules.

External Watchdog Mode

When an external watchdog chip is connected (via GPIO) in 88LX51xx to the WDI pin of the external watchdog, the firmware enables and feeds this external watchdog. If this chip is not fed after a certain time (see Table 62), it produces a power cycle throughout the board.

The following figure shows the external watchdog chip signal chronogram.



Note 1: The WDI pin was a weak pull-up resistor which is disabled after the 1st falling edge on the WDI pin.

Figure 36: External Watchdog Chip Signal Chronogram

Table 62 lists the relevant timing values for the watchdog signals.

Unless otherwise indicated, all electrical specifications limits are specified for V_{DD} = from 1V to 5.5V, R_{PU} = 100kΩ (only MCP1320, MCP1321, and MCP1322), TA = from –40°C to 125°C.

Table 62: Timing Values for Watchdog Signals

Parameter	Sym	Min	Typ	Max	Units	Conditions
WDI Pulse Width	t_{WP}	50	-	-	ns	-
Watchdog Timeout Period	t_{WD}	4.3	6.3	9.3	ms	See Note 1 in Figure 36.
		71	102	153	ms	See Note 1 in Figure 36.
		1.12	1.6	2.4	sec	Standard timeout.
		17.9	25.6	38.4	sec	See Note 1 in Figure 36.

Boot Phase

After a power cycle, the firmware must configure the GPIO high at bootup. Currently, the GPIO is configured as INPUT at bootup and there is a pull-up resistor.

According to the external watchdog chip data model for the MCP132X series (http://ww1.microchip.com/downloads/en/devicedoc/21985d.pdf), the watchdog timer (WDT) is activated by the first falling edge on the WDI pin.

Thus, until the keep-alive task does not produce the first toggle, the external watchdog is idle.

Monitor Phase

Once the firmware boots, it must start the keep-alive task, allowing the sending of pulses through the GPIO.

The following table lists the relation between time values compared to TWD.

Table 63: Relation Between Time Values Compared to TWD

Parameter	Time	Description
TWD	1.12 sec	Default parameter.
Task Timeout	TWD/2	You can reduce this number if necessary.
Pulse Width	TWD/4	A timer performs a toggle in the GPIO.

The pulses must be sent at each timeout until the implemented external conditions decide that the system needs a reset and block the sending of new pulses, so that the external watchdog power cycles the board.

Reset Phase

If the firmware decides to perform a reset due to a watchdog or an *on demand* reset, for example, the system has a second reset due to a power cycle forced by the external watchdog. This behavior is not an infinite loop because the external watchdog needs to be restarted after the reset.

Non-External Watchdog Mode

If there is no external watchdog, the keep-alive activity is reduced to check the device connectivity within the network and if a malfunction is detected, it performs a reset which only affects the 88LX51xx chip, not the whole board.

Table 64: Parameters Related to Keep-Alive Functionality

Parameter	Description
`KEEPALIVE.GENERAL.ENABLED_AT_BOOT`	If set to *YES*, The keep-alive functionality is enabled at bootup The default value is YES.
`KEEPALIVE.GENERAL.CONFIG[8]`	Array of 8 bytes with configuration parameters related to keep-alive checks. It must be redefined for HN.
`KEEPALIVE.GENERAL.EXT_WD`	Configures whether or not an external watchdog is used.
`KEEPALIVE.GENERAL.GPIO`	GPIO connected to the external watchdog (if present).
`KEEPALIVE.GENERAL.TIMEOUT`	Timeout to feed the external watchdog. It must be at least half the timeout to reset the external watchdog or less, but MaxLinear recommends 1/4.
`KEEPALIVE.GENERAL.TIME_TO_RESET`	Countdown which indicates the time remaining until the next reset (<u>for information only</u>).

Corporate Headquarters:
5966 La Place Court, Suite 100
Carlsbad, CA 92008
Tel.: +1 (760) 692-0711
Fax: +1 (760) 444-8598

www.maxlinear.com

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by MaxLinear, Inc. MaxLinear, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced into, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of MaxLinear, Inc.

Maxlinear, Inc. does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless MaxLinear, Inc. receives, in writing, assurances to its satisfaction that: (a) the risk of injury or damage has been minimized; (b) the user assumes all such risks; (c) potential liability of MaxLinear, Inc. is adequately protected under the circumstances.

MaxLinear, Inc. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from MaxLinear, Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

MaxLinear, the MaxLinear logo, any MaxLinear trademarks (MxL, Full-Spectrum Capture, FSC, G.now, AirPHY, Puma, and AnyWAN), and the MaxLinear logo on the products sold are all property of MaxLinear, Inc. or one of MaxLinear's subsidiaries in the U.S.A. and other countries. All rights reserved. Other company trademarks and product names appearing herein are the property of their respective owners.