MAXLINEAR

ENVISIONING • EMPOWERING • EXCELLING

G.hn Spirit P2MP Software

User Guide

Revision History

Document No.	Release Date	Change Description
058UGR00	02/19/18	Initial release.

Table of Contents

List of Figures

List of Tables

Firmware Description

The embedded microprocessor in the Digital Baseband (DBB) processor runs MaxLinear's time-proven Spirit P2MP firmware that provides a rich set of communication and management applications. It also provides a flexible API that enables developers to build their own code. It offers a high degree of customization of existing features.

External processors are only required for specific applications.

The interconnection to external application processors can be achieved using:

- SPI
- SDIO
- MII
- UART interfaces

The Spirit Point-to-Multipoint (P2MP) firmware manages how the G.hn nodes intercommunicate. The G.hn nodes support coaxial in-building scenarios where multiple end points (CPEs) share the same coaxial distribution network. Special MAC schemes have been implemented to maximize the efficiency of the network.

The Spirit P2MP firmware is composed of several independent modules, which can be added or removed depending on the customer's specific needs.

Capabilities

Table 1: List of Capabilities Supported by Spirit P2MP 7.6 G.hn

Capability	Maximum Value
Number of supported nodes in a network	17 nodes (1 HE + 16 CPEs)
Number of connections per node	Separated connections: ■ 1 for user data ■ 1 for G.hn management
Ethernet MAC addresses per G.hn domain	1024
FEC payload size	■ 120 bytes for broadcast connection ■ 540 bytes for data and management
FEC rates	1/2, 3/4, 5/6, and 16/18
Number of simultaneous multicast channels routed	128

Note: This document contains references to configuration parameters. The complete information about the configuration parameters can be found in the Spirit_P2MP_v7_6_xxx_Docs package delivered with the firmware.

Firmware Core Features

Real Time Operating System and POSIX Interface Core

The embedded firmware runs on top of a customized uCOS-II kernel. The benefit of using a Real-Time Operating System (RTOS) is that it enables a low latency rate and provides greater stability.

RTOS is an operating system that enables multiple programs to run at the same time. This is called multitasking. The RTOS provides a preemptive and deterministic multitask environment that includes features such as interrupt handling, semaphores, event flags, tasks, time, timer management, message mailboxes, and queues.

The RTOS services are not directly exposed to the Software Development Kit (SDK) developer. Instead, a POSIX compliant layer sits on the top of the RTOS, which facilitates the porting of existing applications.

Flash File System

The Spirit P2MP firmware uses a flash file system to manage the flash memory space and to store firmware images, configuration files, and other types of files that are required for applications such as log captures and web servers.

The file system:

- Supports long names (up to 128 characters), directories, and concurrent access.
- Provides Fail Safe (FS) access to the flash memory.

  Fail Safe means that it is robust in front of power cuts or resets during write operations in flash.

  There are two copies of the descriptor table file, so that the consistency of the file's system is preserved in the event of power interruption. This prevents files from becoming corrupted if there is a problem during writing operations.
- Implements a wear leveling algorithm to equalize the use of the sectors and prevent an early failure of a frequently accessed sector.
- Offers POSIX-standard functions to manipulate files and facilitate access to the flash memory from the API.

TCP/IP Stack

The Spirit P2MP firmware includes a TCP/IPv4 and IPv6 dual stack that supports the following protocols:

- IP
- UDP
- TCP
- ARP
- ICMP

The stack itself is not needed for basic G.hn transceiver functions. However, it can be used for remote accessibility purposes, such as operation, maintenance, and configuration.

The UDP/TCP protocols allow you to create sockets with other IP machines using high-level protocols such as FTP, and HTTP.

The TCP/IP stack supports IP fragmentation.

Trivial File Transfer Protocol Client

The firmware includes a Trivial File Transfer Protocol (TFTP) client to transfer files to and from a TFTP server. It is used to download new firmware versions into the nodes.

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

File Transfer Protocol Client

The firmware includes a File Transfer Protocol (FTP) client to transfer files to and from an FTP server. It is used to download new firmware versions into the nodes.

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

Domain Name System Client

A Domain Name System (DNS) client is incorporated into the firmware. The DNS clients are referred to as resolvers in 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, such as the Network Time Protocol (NTP) server, by using the DNS URLs.

Table 2: DNS Configuration Parameters

Parameter	Value
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 (NTP) Client

A Network Time Protocol (NTP) client is incorporated into the firmware.

This protocol enables a Real Time Clock (RTC) in the system.

Table 3: NTP Configuration Parameters

Parameter	Value
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	Enable or Disable the NTP client.
NTP.GENERAL.RESYNC_TIME	Configure the re-synchronization interval time (in minutes).

Table 3: NTP Configuration Parameters (Continued)

Parameter	Value
NTP.GENERAL.STATUS	NTP statuses: ■ Amount of disabled clients. ■ Unsynchronized clients (the absolute time has not been set yet). ■ Synchronized clients (the device has acquired accurate absolute time). ■ Error_FailedToSynchronize (the device failed to acquire accurate absolute time).

Dynamic Host Configuration Protocol Client

The Spirit P2MP firmware includes a Dynamic Host Configuration Protocol (DHCP) client to automatically configure the basic IP parameters (IP address, subnet, mask, default gateway address, option-82, and option-125) from a DHCP server in the network.

HTTP Server

The Spirit P2MP firmware includes a HTTP server that implements a subset of the *RFC 2616 (HTTP/1.1)* and provides an easy-to-use API for common HTTP items, such as headers and request parameters. Embedded systems optimize the firmware's architecture and enable it to run efficiently on systems with limited resources.

Configurable Flash Support

The Spirit P2MP firmware provides several flash devices. Customers can select different flash devices and develop their own flash device drivers based on ones that are already available.

Support for the Loader Module

The Spirit P2MP firmware implements a loader module which initializes:

- The DBB processor, and any external components.
- The required hardware blocks that are needed to obtain access to the DDR memory, and other interfaces. It does this by:
  - Locating the firmware image in the flash file system.
  - Reading and unzipping the firmware images from the flash into the RAM.
  - Starting the execution of the firmware after the device initialization is complete.

There are two slots in the flash that are used to store two images of the module loader and to ensure that the module loader can be remotely upgraded in a secure manner.

System Boot Process

The first code that the G.P2MP modem executes is the ROM boot code. It configures the Ethernet interfaces and some other hardware modules so that it can access the external flash memory. It then examines the flash memory for a valid binary (the module loader) and loads it.

The loader module is then read from the flash memory into SRAM and it is executed. It performs additional configurations such as clocks and external DDR memory.

The loader can:

- Access the flash file system and locate the main firmware file.
- Read the firmware file from the flash.
- Unzip the firmware file.
- Load the firmware file into the suitable memory sections.

After the operation is finished, the main firmware code is executed.

If the boot process fails, the device enters into Recovery Mode.

Recovery Mode

If the flash memory is empty or corrupted, or if the G.P2MP node is powered on while pushing the **CONFIG** button, the flash memory boots up through the Ethernet port. It sends `ROM Boot` broadcast packets and waits for a binary file to load in the memory through the Ethernet interface. The Ethernet interface does not use an Ethernet protocol by itself, it uses the recovery mode. Once the firmware is loaded in the modem, it can be flashed using the flash upgrade from the Spirit Configuration Tool (SCT).

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

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 the:

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

The parameters can be set in two different ways:

- Using the PCK: When generating a complete FLASH image.
- Using the PTK: When executing the production test.

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

Secure Upgrade

A secure upgrade (Fail Safe) of any flash binary sections or files included in the Spirit firmware release is performed by backing up images during the upgrade procedure. You can upgrade the following:

- The firmware
- The loader module
- Configuration files
- Factory reset configuration

The upgrades can be done using Layer 3 (FTP, TFTP) or Layer 2 (L2Upgrade) protocols. MaxLinear only recommends an L2Upgrade for G.P2MP nodes that are connected locally through an Ethernet network.

The upgrade can also be used to update any other files that are included in the flash FS. This can be beneficial for customer specific applications such as web server customizations.

Factory Reset

The Factory Reset operation:

- Recovers the configuration stored in the factory reset configuration file and applies it to the modems. The configuration file can be customized with the PCK that was defined in production, although you can also upgrade it remotely.
- Overwrites any settings that could have been 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 that are 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 button GPIO and the time).

Ethernet Driver

The Spirit P2MP firmware provides support to connect one Ethernet device to the DBB processor though a RGMII or SGMII interface.

You can define the Ethernet interface type and its parameters (mode, IFG, delays, and MIIM mode) by Using the `ETHIFDRIVER` parameter.

You should take into account the following limitations:

- The RGMII interface can only be connected to the ETHA port.
- The SGMII interface can only be connected only to the ETHB port.
- Only one Ethernet interface can be enabled at the same time.

You can define the Ethernet link capabilities, duplex option, and check the status of the link by Using the ETHPHYCONF parameters

You should take into account the following limitations:

- MaxLinear does not recommend disabling auto-negotiation. If it is disabled, you can modify the configured speed by using ETHPHYCONF parameters.
- MaxLinear recommends using auto-negotiation to force a specific speed and duplex configuration. You should only enable the desired capabilities, and disable the others.
- Half-duplex is not supported.

Watchdog Module

The Spirit P2MP firmware implements a watchdog module to monitor the correct operation of the firmware. The watchdog module continuously ensures that:

- There is no memory corruption.
- The RTOS is operating properly.
- There are no problems in any functional modules in the adapter.

G.hn PHY

PHY Layer Support

The Spirit P2MP firmware includes support of the physical layer G.hn specifications:

- *ITU-T G.9960* defines the core or the physical layer for G.hn systems.
- *ITU-T G.9964* states the requirements of Power Spectral Density (PSD) that the G.hn system needs to meet.

The following subsections provide additional details about the level of compliance needed:

- Physical medium attachment sublayer support as described in G.9960, subclause 7.1.3.
- Physical medium dependent sublayer for a PLC mode as described in G.9960, subclause 7.1.4.
- Predefined and runtime Bit Allocation Table (BAT) support as described in G.9960, subclause 7.1.4.2.2.
- Predefined and runtime Bit and Tx Port Mapping Allocation Table (BMAT) support as described in G.9963, subclause 7.1.4.4.3.
- Automatic selection of Robust Communication Mode (RCM) transmission mode as described in G.9960, subclause 7.1.3.3.
- PHY frame formats as described in G 9960, subclause 7.1.2.1.

Profiles Support

The Spirit P2MP firmware includes support for the G.9964 profiles listed in the following table.

Table 4: Profile in Spirit P2MP

Bandplan	Media	Type	Notes
200MHz	Coax	SISO	Meets the PSD requirements of the G.9964 subclause 6.3.2, injecting a PSD limited to –76dBm/Hz.

PSD and Notching Configuration

The Spirit P2MP firmware allows the definition of new notches and a PSD modification:

- Vendor Power Mask: A set of notches that can be defined by the product manufacturer through the PCK. The notches can be enabled or disabled. For further information, refer to the help embedded in the PCK tool delivered with the firmware.
- User Power Mask: A set of notches that can be defined by the user through any of the provided configuration methods, such as the SCT. For further information, refer to the help embedded in the SCT delivered with the firmware.

G.hn MAC and QoS

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

The DBB processor based devices running Spirit P2MP firmware support both Head End (HE) and Customer Premise Equipment (CPE) roles.

The Spirit P2MP firmware includes support for most of the features described in ITU G.hn G.9961. Support for the firmware is described in the following sections.

Based on *Homegrid Forum* (HGF) recommendations, some extra functions are provided in addition to the features described in ITU G.hn G.9961.

Head End Functions

The Spirit P2MP HE firmware implements a subset of the Domain Master (DM) functionalities described in G.9961, subclause 8.6. The functions are:

- Network admission protocol.
- Bandwidth management. The Spirit P2MP firmware implements priority-based scheduling schema, by giving transmission time in the network to and from each CPE for the downstream and the upstream flows.
- Routing and topology management.

Customer Premises Equipment Functions

The Spirit P2MP firmware implements the following End Point (EP) functionalities as described in *ITU-T G.9961*, subclause 8.5. The functions are:

- MAC cycle synchronization and synchronized transmissions. The CPE synchronizes its MAC cycle and MAC clock with the HE and follows the TXOP and time slot assignments in the MAC cycle.
- The routing of Application Data Primitives (ADPs).
- The broadcasting of LLC frames.
- Retransmissions and Layer 2 acknowledgments.

Medium Access Control

The Spirit P2MP firmware implements a TDMA-based medium access control that is synchronized with an internal synchronization signal every 40ms.

For a more detailed description about the MAC sub-layer, refer to the specifications listed in *ITU-T G.9960.*

The Spirit P2MP firmware uses a Contention Free Transmission Opportunity (CFTXOP) for the HE that is used to start a bidirectional transmission with the CPE.

The Spirit P2MP firmware implements a MAC scheme using the CFTXOP that is assigned to the HE that is divided by the HE into slots in a scheduling list. The slots are allocated for each CPE. For each slot there is a downstream and an upstream traffic exchange that uses the bidirectional transmission mechanism as specified in *ITU-T G.9961*.

The following figure displays the scheduling list of allocated slots. Each slot is assigned to one CPE.

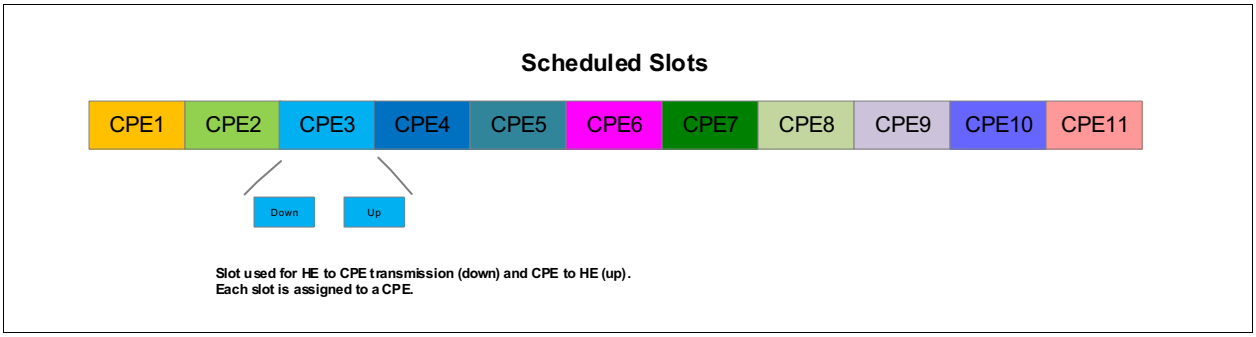

Figure 1: Slots Allocation in Spirit P2MP 7.6 MAC

The slot is used for downstream (HE to CPE) and upstream transmissions (CPE to HE).

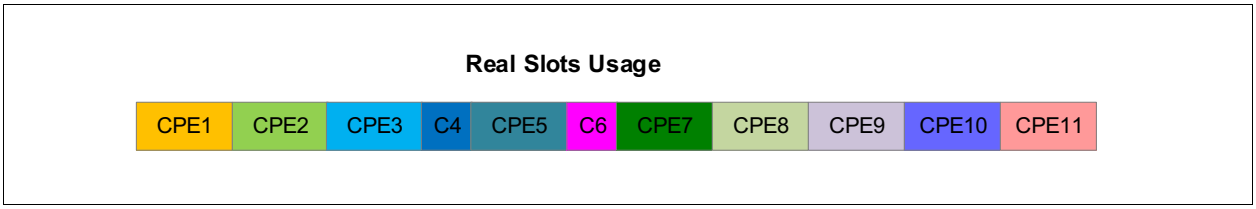

Figure 2: Slots Usage in Spirit P2MP 7.6 MAC

When one slot finishes, the next slot starts. This results in improved latency and overall efficiency.

QoS

The Spirit P2MP firmware implements:

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

The HE implements QoS policies to define how the MAC layer distributes the time slots among different nodes. This improves the efficiency in point-to-multipoint networks by implementing media access policies among network nodes depending on the priority of their traffic and activity.

QoS Service Profiles

A QoS service profile is a group of traffic types that share the same properties of throughput and latency.

The QoS engine guarantees the same latency policy for flows that share the same service profile.

There are four different services sorted from 1–4, with one being the lowest priority, and four being the highest priority.

The following table lists the different service profiles and their correspondence with 802.1p (VLAN), DSCP, and TOS priorities.

Table 5: QoS Service Profiles 802.1p, DSCP, and ToS

Service	Properties	802.1p	DSCP	ToS	Traffic Example
Service 1	High Xput + High Latency	0, 2	0x0–0xF	0, 1	Data
Service 2	High Xput + Mid Latency	1, 3	0x10–0x1F	2, 3	Prioritized data, TCP ACKs
Service 3	Mid Xput + Mid Latency	4, 5	0x20–0x2F	4, 5	Video
Service 4	Low Xput + Low Latency	6, 7	0x30–0x3F	6, 7	VoIP and Management

Buffer Management Policies

In P2MP scenarios with a large amount of CPEs, managing packet buffers is essential to maintain the QoS when the network is congested.

The P2MP firmware takes advantage of the 88LX5153 G.hn DBB processor by including an algorithm to optimize the packet buffer.

In the 88LX5153 there are two types of packet buffers that can be assigned on each connection:

- CBR buffers: Committed buffers. 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. Memory is assigned to a connection that can be shared with other connections that have been 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 what type of buffer it is.

- CBR buffers: Defined by a policy that divides the CBR buffer pool between the number of established data connections. It is controlled by the parameter `QOS.QUEUEMANAGEMENT.CBR_POLICY`. The values allowed are *EVENLY* and *BWLIMIT*.
- VBR buffers: Defined by a policy that assigns a percentage of the total VBR buffer pool to each established data connection. It is controlled by the parameter `QOS.QUEUEMANAGEMENT.SATURATION_POLICY`. The values allowed are *MAX_MEMORY_USE* and *BWLIMIT*. The parameter `QOS.QUEUEMANAGEMENT.MAX_MEMORY_USE` determines the percentage amount of the total VBR pool that is assigned to each data connection.

The corresponding configuration parameters are shown in the following table.

Table 6: Buffer Management Configuration Parameters

Parameter	Value
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 ■ BWLIMIT
QOS.QUEUEMANAGEMENT.MAX_MEMORY_USE	The maximum percentage amount of the total buffer memory for packets in one data connection.
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 ■ BWLIMIT

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

Bandwidth Limitation

Bandwidth limitation limits the average amount of traffic that can be transmitted and received by a CPE. Bandwidth can be limited in both upstream and downstream links.

You can define up to 16 different bandwidth limitation profiles. You can assign each profile to one or more CPEs, associated MACs, and Profile IDs, as described in Table 7 on page 13.

Bandwidth limitation profiles can only be provisioned in the HE of the network.

For every CPE registering the network:

- The HE detects the profile to apply by using the PROFILE_ID table or the DEFAULT_PROFILE. It uses the DEFAULT_PROFILE when the CPE MAC address is not included in the table.
- The HE remotely applies the profile bandwidth limitation settings on the corresponding CPE.

The profile applied in the CPE is shown in the parameter `QOSENGINE.USERPROFILEMAP.CPE_PROFILE.` This parameter stores the CPE profile that it receives from the HE. This parameter is not configurable. It is for informational purposes only.

The bandwidth limitation configuration parameters only apply to the HE. They are listed in the following table.

Table 7: Bandwidth Limitation Configuration Parameters

Parameter	Value
QOSENGINE.USERPROFILEMAP.MAC	List of MACs with defined profiles. A maximum of 16 entries can be defined.
QOSENGINE.USERPROFILEMAP.PROFILE_ID	Profile ID that is associated to a user. The Profile ID in the entry X of this list is associated with the user with the MAC parameter QOSENGINE.USERPROFILEMAP.MAC.X.
QOSENGINE.USERPROFILEMAP.DEFAULT_PROFILE	Profile used by the CPEs with a MAC not defined in the QOSENGINE.USERPROFILEMAP.MAC parameter.
QOSENGINE.USERPROFILES.BWLIMIT_ENABLED	Enable or Disable bandwidth limit in one profile. This parameter applies to both upstream and downstream links.
QOSENGINE.USERPROFILES.BWLIMIT_UP_MAXRATE	Maximum upstream rate from the CPE ■ Minimum value = 10Mbps ■ Maximum value = 800Mbps
QOSENGINE.USERPROFILES.BWLIMIT_DOWN_MAXRATE	Maximum downstream rate to the CPE ■ Minimum value = 10Mbps ■ Maximum value = 800Mbps
QOSENGINE.USERPROFILES.ALLOWED_SERVICE	This list of parameters determines if a QoS is allowed in a user profile. The service latency is not guaranteed unless the service is allowed in this table (even if service priority is detected in the user traffic). This is a BOOLEAN array consisting of four elements where services 1–4 go from entry 1–4.
QOSENGINE.USERPROFILES.BWLIMIT_ON_SERVICE	This list of parameters determines if a QoS is included in bandwidth limitation. This is a BOOLEAN array consisting of four elements where services 1–4 go from entry 1–4.

Configuring Bandwidth Limitation example

The following figure shows four CPEs and their requirements:

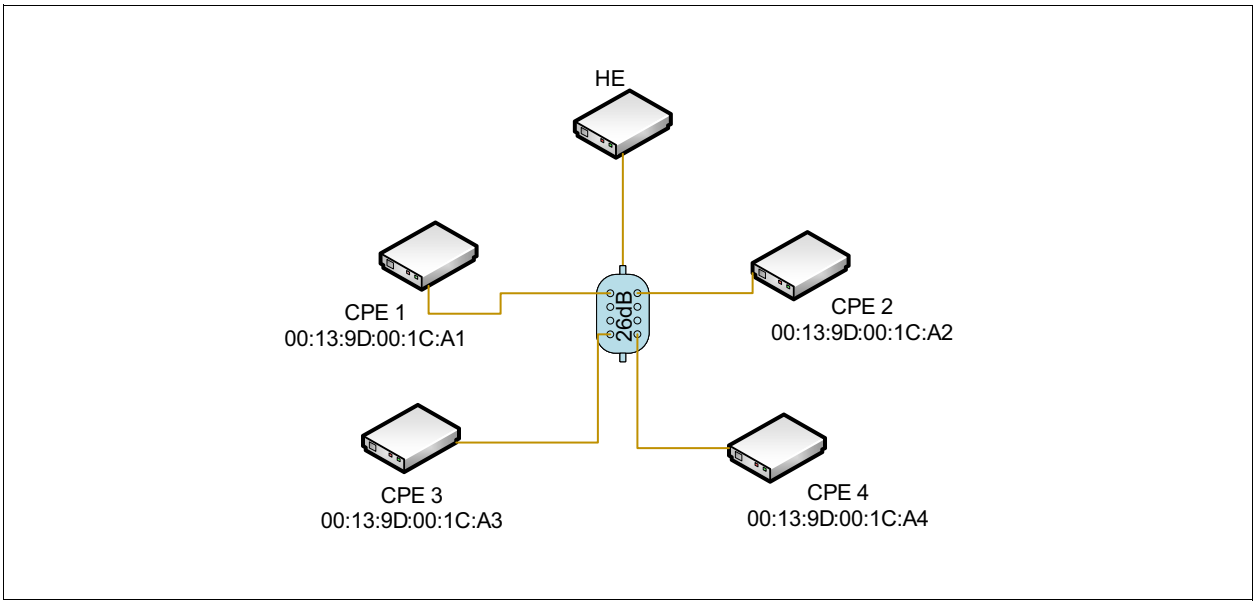

Figure 3: Bandwidth Limitation Example

Table 8: Bandwidth Limitation Requirements Example

CPE	Downstream Maximum Rate (Mbps)	Upstream Maximum Rate (Mbps)	Service Profiles Included in Bandwidth Limitation
1	50	10	All
2	70	15	1, 2, 3
3	50	10	All
4	80	20	1, 2

Note:

- CPE1 and CPE3 have the same requirements. They can use the same profile, which is defined as Profile 1 in this example.
- CPE2 uses Profile 2.
- CPE4 uses Profile 3.

First, define the profiles and their properties.

Table 9: Profiles Definition Example

Profile	Parameter	Value
1	QOSENGINE.USERPROFILES.BWLIMIT_DOWN_MAXRATE.1	50
1	QOSENGINE.USERPROFILES.BWLIMIT_UP_MAXRATE.1	10
1	QOSENGINE.USERPROFILES.BWLIMIT_ON_SERVICE.1.0	YES, YES, YES, YES
1	QOSENGINE.USERPROFILES.BWLIMIT_ENABLED.1	YES
1	QOSENGINE.USERPROFILES.BWLIMIT_ENABLED.1	YES
2	QOSENGINE.USERPROFILES.BWLIMIT_DOWN_MAXRATE.2	70
2	QOSENGINE.USERPROFILES.BWLIMIT_UP_MAXRATE.2	15
2	QOSENGINE.USERPROFILES.BWLIMIT_ON_SERVICE.2.0	YES, YES, YES, NO
2	QOSENGINE.USERPROFILES.BWLIMIT_ENABLED.2	YES
3	QOSENGINE.USERPROFILES.BWLIMIT_DOWN_MAXRATE.3	80
3	QOSENGINE.USERPROFILES.BWLIMIT_UP_MAXRATE.3	20
3	QOSENGINE.USERPROFILES.BWLIMIT_ON_SERVICE.3.0	YES, YES, NO, NO
3	QOSENGINE.USERPROFILES.BWLIMIT_ENABLED.3	YES

Second, define the list of the CPEs.

Table 10: CPEs Definition Example

Parameter	Value
QOSENGINE.USERPROFILEMAP.MAC.1	00:13:9D:00:1C:A1
QOSENGINE.USERPROFILEMAP.MAC.2	00:13:9D:00:1C:A2
QOSENGINE.USERPROFILEMAP.MAC.3	00:13:9D:00:1C:A3
QOSENGINE.USERPROFILEMAP.MAC.4	00:13:9D:00:1C:A4

Finally, define the association between each CPE and their profile.

Table 11: Association Between CPEs and Profiles Example

Parameter	Value
QOSENGINE.USERPROFILEMAP.PROFILE_ID.1	1
QOSENGINE.USERPROFILEMAP.PROFILE_ID.2	2
QOSENGINE.USERPROFILEMAP.PROFILE_ID.3	1
QOSENGINE.USERPROFILEMAP.PROFILE_ID.4	3

Traffic Awareness

Traffic awareness is used to optimize the performance of the network by reducing the time slot allocations for the CPEs which do not require it. The algorithm determines when a CPE is IDLE or ACTIVE based on the traffic rate between the CPE and the HE.

When a node is IDLE, the transmitting and receiving time slot allocations for the CPE are reduced, and the latency is increased to 80ms.

If traffic above a threshold is detected, the node becomes ACTIVE and time slot allocations are assigned to the CPE with a maximum latency of 20ms.

Service profiles use the following criteria to determine if a node is IDLE or ACTIVE:

- For traffic that belongs to Service Profiles 1 and 2, the CPE is ACTIVE if the throughput is above what is listed in the `QOS.SCHED.TRAFFIC_AWARE_XPUT_THR` parameter (in Kbps). The default value is 1Mbps.
- For traffic that belongs to Service Profiles 3 and 4, the node is considered active despite the throughput if they are allowed for a given CPE. See parameter `QOSENGINE.USERPROFILES.ALLOWED_SERVICE` for more information.

Data Link Layer

Network Admission Protocol

The network admission protocol is used to:

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

The protocol also performs periodic re-registrations to keep track of the nodes that are present in the domain. The alloted time is configurable and the HE decides its value.

The Spirit P2MP firmware implements the network admission protocol as specified in the *G.9961 Corrigendum 2, subclause 8.6.1*.

Access Control List

The access control list:

- Determines which CPEs can register into the network by checking the MAC address of the CPE. It is based on a MAC address list of authorized CPEs. If the MAC address of the CPE trying to register into the network is not in the list, the CPE is not allowed to register into the domain.
- Displays the CPEs that are trying to register into the network. This list facilitates the provisioning task of the new CPEs, and the fraud detection for non-provisioned CPEs.
- Implements a new parameter to enable or disable access control.

Table 12: Access Control List Parameters

Parameter	Value
NAP.GENERAL.ACCESS_CONTROL_ENABLED	Enable or Disable access control
NAP.GENERAL.ACCESS_REQUEST_MAC_LIST	List of the MACs of the CPEs trying to register but are not allowed. Maximum size = 32 CPEs.
NAP.GENERAL.ACCESS_ALLOWED_MAC_LIST	List of the MACs of the CPEs authorized to register to the HE. This parameter replaces the NAP.GENERAL.MAC_ACCESS_LIST parameter. Maximum size = 16 CPEs.

Blacklist for MAC Addresses

The blacklist for MAC addresses can:

- Block the traffic from a given source MAC address.
- Filter a maximum of eight source MAC addresses that work as a blacklist for those MAC addresses.

The blacklist is only enabled in CPEs. The traffic is filtered in the CPE's Ethernet interface, and only affects incoming traffic.

Table 13: MAC Blacklist Parameters

Parameter	Value
BFT.BLACKLIST.ENABLED	Enable or Disable the MAC source address blacklisting feature.
BFT.BLACKLIST.MACS	Table containing the source MAC addresses that are blacklisted. Maximum size = Eight MAC addresses.

Note: You must have the `RTM.GENERAL.UNKNOWN_SUPPRESSION` parameter enabled.

Limit the Maximum Number of MAC Addresses

Using the Spirit P2MP firmware, you can limit the maximum number of MAC addresses a CPE can learn through its Ethernet interface. This enables you to restrict the number of devices that are connected to the CPE's Ethernet.

This feature is only enabled in CPEs. The CPE learns the limit of configured MAC addresses. After the limit is reached, it stops learning new MAC addresses. If the MAC addresses are deleted due to ageing and the number of learned MAC addresses is below the limit, the CPEs are then able to learn new MAC addresses.

Table 14: Limit the Maximum Number of MAC Address Parameters

Parameter	Value
BFT.GENERAL.LOCAL_MACS_LIMIT_ENABLED	Enable the limit of MAC addresses that can be learned from the Ethernet port.
BFT.GENERAL.LOCAL_MACS_LIMIT	Maximum number of MACs that can be learned from the Ethernet port. ■ Minimum = 1 MAC ■ Maximum = The maximum number of MAC addresses supported (see "Capabilities" on page 1).

Note: You must have the RTM.GENERAL.UNKNOWN_SUPPRESSION parameter enabled.

Channel Estimation Protocol

The Channel Estimation Protocol (CEP) is used to measure the characteristics of the channel between the transmitter (source) and the receiver (destination) nodes.

The procedure involves:

- Monitoring the channel condition.
- Transmitting the PROBE frames.
- Selecting the best modulation scheme.
- The exchange of:
  - BAT for SISO
  - FEC coding rate
  - Guard interval for payload

The Spirit P2MP firmware supports CPEs as described in *ITU-T G.9961* for SISO modes.

CPE Isolation

The Spirit P2MP firmware uses a Routing and Topology Maintenance Protocol (RTM) as specified in *ITU-T* G.9961*, subclauses 8.5.3, 8.5.4, and 8.6.4.*

The implementation of the RTM has been optimized for EoC/MDU scenarios to guarantee the traffic isolation between CPEs. This prevents direct connections between CPEs (peer-to-peer), and creates forward and backward traffic routes from the CPE to the HE.

Broadcast Suppression

This functionality limits broadcast input data rate in Ethernet port to a maximum value, dropping any exceeding packets that are received.

Unknown Traffic Suppression

This functionality drops all input traffic, both unicast and multicast, in the Ethernet ports with a destination MAC address that has not been solved.

The Spirit P2MP firmware enables this feature by default.

Traffic Statistics

The Spirit P2MP firmware provides a complete set of traffic statistics.

Table 15: Traffic Statistics Parameters

Parameter	Value
ETHIFDRIVER.STATS.INFO_DESC	Description of Ethernet traffic counters.
ETHIFDRIVER.STATS.INFO	Ethernet traffic counters.
ETHIFDRIVER.STATS.RESET	Global reset for all Ethernet traffic counters.
QOS.STATS.G9962_DESC	*ITU-T G.9962* counters description.
QOS.STATS.G9962	*ITU-T G.9962* counters.
QOS.STATS.RESET	Global reset for *ITU-T G.9962* counters.

Traffic Shaping

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

The traffic shaping is important for P2MP CPE devices that are connected to low speed devices with 100Mbps Ethernet interfaces because the CPEs can receive data on their G.hn interfaces at 1.6Gbps.

The following table list the configuration parameter that is used to enable traffic shaping.

Table 16: Traffic Shaping Configuration Parameter

Parameter	Value
FLOWMONITOR.TRAFFSHAP.ENABLED	YES or NO Default = YES

This functionality is not activated in the HE because:

- The HE receives traffic from all the CPEs and conforming the traffic would affect the reception from other CPEs.
- The HE should have a high speed uplink connection (1Gbps or 2.5Gbps Ethernet port) and it does not require traffic shaping.

However, for scenarios where the HE Ethernet speed is 2.5Gbps and the uplink is up to 1Gbps (that is 1G EPON or GPON), the maximum rate at which the HE outputs the packets in the upstream should be limited.

The following table lists the configuration parameters used to limit the maximum Ethernet output.

Table 17: Forced Rate Configuration Parameter

Parameter	Value
FLOWMONITOR.TRAFFSHAP.ENABLED	YES or NO Default = YES
FLOWMONITOR.TRAFFSHAP.FORCED_RATE	Rate in Mbps. (A zero value means disabled).

Caution: Traffic shaping should be used in CPEs. MaxLinear recommends that you force the rate configuration in HE when the Ethernet speed cannot be changed (that is, it requires a hardware modification).

VLAN

IEEE 802.1Q is part of the IEEE 802.1D standard. It defines a system of Virtual LAN (VLAN) tagging for Ethernet frames by adding a header to Ethernet frames and by specifying its use by bridges and switches

The Spirit P2MP firmware partially supports IEEE 802.1Q by implementing the tagging, untagging, and filtering capabilities on the Ethernet and management interfaces.

Tagging

Using the Spirit P2MP firmware, you can configure a VLAN tag. You can remove the VLAN tag for outgoing Ethernet frames. The same behavior can be applied to the management interface.

The tagging behavior can be defined on each interface as:

- ACCESS
  - Ingress packets. Tagged with the specified VLAN tag for this interface.
  - Egress packets. The VLAN tag is removed.
- TRUNK
  - A VLAN tag is specified as PVID in a trunk interface.
    - Ingress packets without a VLAN tag are tagged with the specific tag.
    - If an egress packet has this VLAN tag, the VLAN tag is removed. Otherwise the VLAN tag is unchanged,
  - PVID is not specified in a trunk interface (PVID = zero). The packets are not modified.
- NONE: The packets are not modified.

Filtering

The Spirit P2MP firmware implements a VLAN filtering function that is based on a set of allowed VLAN tags. Filtering is applied on ingress and egress packets on each interface.

The set of tags can be defined for each individual interface. The maximum number of VLAN tags is 16.

The filtering behavior on each interface depends on its type:

- ACCESS: Only the VLAN tag specified for the interface is allowed. Packets with a VLAN tag that is different than the one specified for the interface are dropped.
- TRUNK: Packets with a VLAN tag are allowed if the tag is in the allowed list of tags.

Caution: When VLAN filtering is active and the interface is configured as NONE, all of the packets on this interface are dropped.

Table 18: VLAN Configuration Parameters

Parameter	Value
VLAN.CVLAN.ENABLE	Activate or Deactivate VLAN (IEEE 802.1Q).
VLAN.CVLAN.FILTERING_ENABLE	Enable or Disable VLAN ingress and egress filtering.
VLAN.CVLAN.PVID_ETHA	VLAN identifier for Ethernet A interface (if it is set to 0, tagging is deactivated).
VLAN.CVLAN.PVID_ETHB	VLAN identifier for Ethernet B interface (if it is set to 0, tagging is deactivated).
VLAN.CVLAN.PVID_MGMT	VLAN identifier for management interface (if it is set to 0, tagging is deactivated).
VLAN.CVLAN.PVID_SDIO	VLAN identifier for SDIO interface (if it is set to 0, tagging is deactivated).
VLAN.CVLAN.CONFIG_IF_ETHA	Port configuration for Ethernet A interface (ACCESS, TRUNK, NONE).
VLAN.CVLAN.CONFIG_IF_ETHB	Port configuration for Ethernet B interface (ACCESS, TRUNK, NONE).
VLAN.CVLAN.CONFIG_IF_MGMT	Port configuration for management interface (ACCESS, TRUNK, NONE).
VLAN.CVLAN.CONFIG_IF_SDIO	Port configuration for SDIO interface (ACCESS, TRUNK, NONE).
VLAN.CVLAN.ALLOWED_TAGS_IN_ETHA	Tags allowed on Ethernet A interface.
VLAN.CVLAN.ALLOWED_TAGS_IN_ETHB	Tags allowed on Ethernet B interface.
VLAN.CVLAN.ALLOWED_TAGS_IN_FW	Tags allowed on management interface.
VLAN.CVLAN.ALLOWED_TAGS_IN_SDIO	Tags allowed on SDIO interface.

Multicast Support

In Internet Protocol Television (IPTV) networks, the P2MP 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 traffic flows entering the Spirit P2MP network are only routed to users who specifically need that traffic flow.

IGMP and MLD Snooping

To efficiently route multicast traffic flows, the Spirit P2MP firmware can snoop on both IGMP and MLD protocol packets. This means:

- 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, both IGMP and MLD snooping are enabled in P2MP products.

Because IGMP works on IP addresses and the Spirit P2MP devices are based on MAC addresses, a mapping of multicast IP address to a destination host MAC is performed.

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

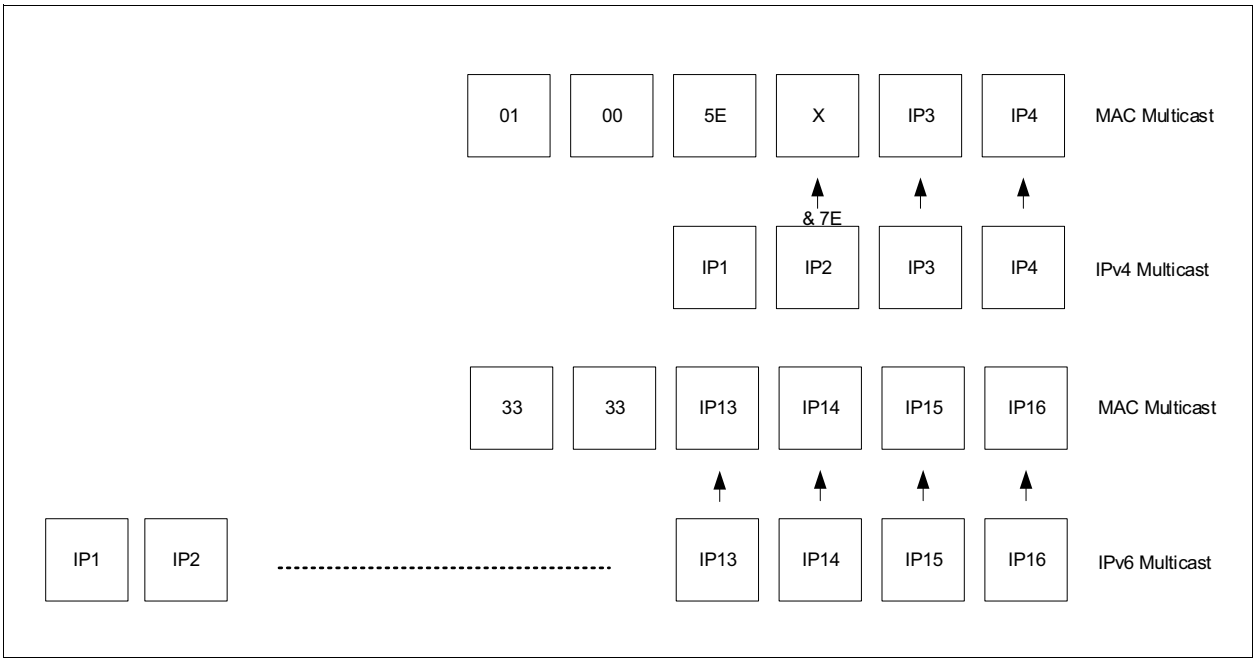

Figure 4: IPv4 to MAC Conversion

Note: Some multicast IP addresses share the same MAC address. Clients 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 P2MP node that is closest to the video server (typically the HE).

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 P2MP network can be configured to:

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

  Note: A multicast set up should 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 P2MP nodes. Specific queries are sent only to nodes having 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.

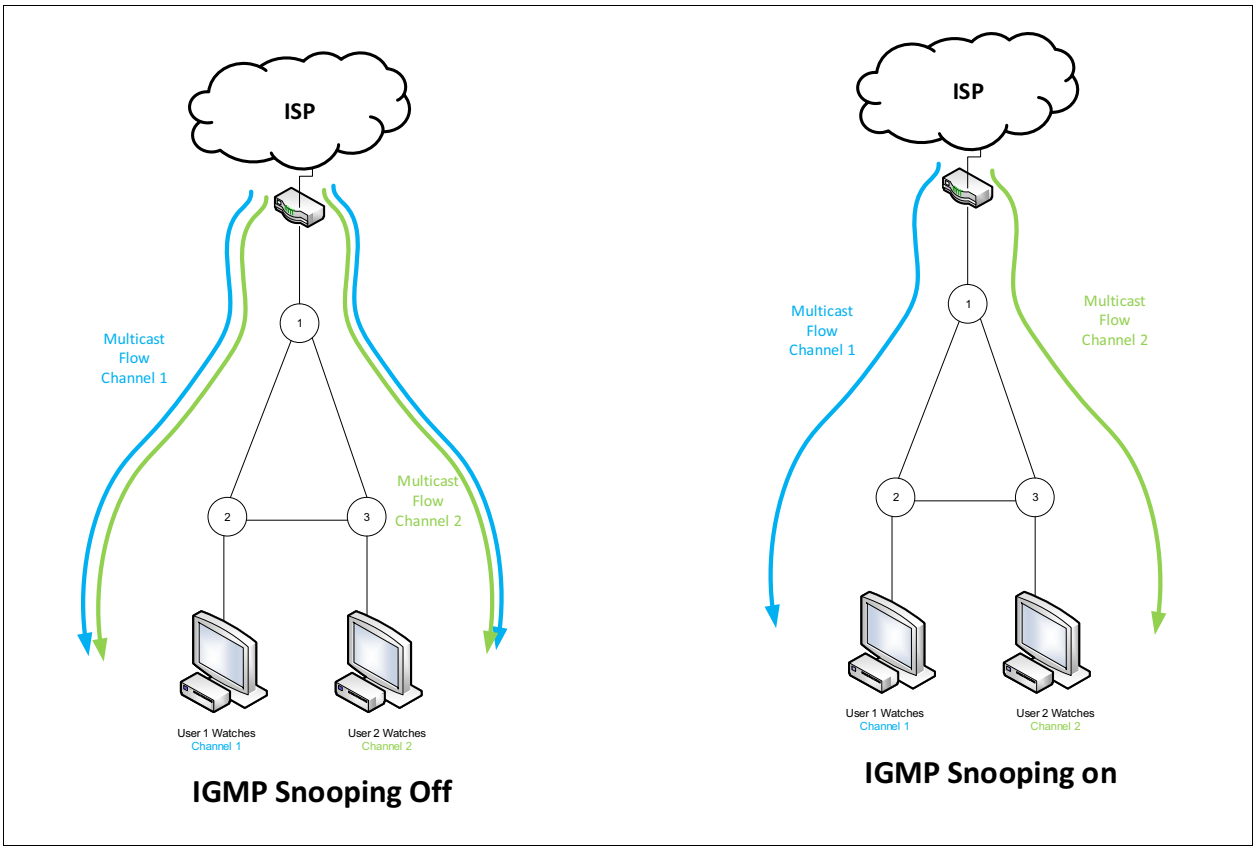

Figure 5: Benefits of IGMP Snooping

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. IGMP packets and their associated multicast data flows outside of 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.

IGMP and MLD Fast Leave

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

This feature is enabled by default.

The following table lists the corresponding configuration parameter.

Table 19: Multicast Fast Leave Configuration Parameter

Parameter	Value
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 Video Source Mode

The Spirit P2MP firmware can control the behavior of a node when it receives an IGMP or MLD query packet. This behavior is configured with the parameter that is described in the following table.

Table 20: Multicast Video Source Configuration Parameter

Parameter	Value
MCAST_GENERAL_VIDEO_SOURCE_MODE	If the value is, ■ AUTO: Detects if the video source is reachable through its Ethernet or G.hn. — Ethernet. Queries are replicated on G.hn to all G.hn nodes. — G.hn. Queries are forwarded to the Ethernet. ■ FORCED: Disables the auto-detection of the video source. — Queries received from G.hn are ignored. — Queries received from Ethernet are replicated to all the G.hn nodes. ■ FORBIDDEN: Disables the auto-detection of the video source. Queries from the Ethernet are ignored. The node never becomes a video source and the queries are never forwarded to other G.hn nodes.

The default values in P2MP products are:

- FORCED in HE devices.
- FORBIDDEN in CPE devices.

Multicast Summary

The G.hn Spirit P2MP firmware 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 *RFC 5790 Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) protocols*. Any report packages that display `IS_IN {x}, TO_IN {x}, ALLOW{x}, IS_EX { }` are considered a *REPORT* message. Other message are considered *LEAVE* messages.
- Source filtering is not supported.

Traffic Prioritization

The Spirit P2MP firmware implements traffic prioritization based on:

- IEEE 802.1p
- DSCP
- User-defined rules

The Spirit P2MP firmware also implements a predefined set of traffic prioritization rules for commonly found traffic, such as, ARP and TCP IPv4 and IPv6.

Prioritization Rules Order

Using the Spirit P2MP firmware, you can define the priority order between IEEE 802.1p and DSCP.

If you enable custom and predefined rules for commonly found traffic, the Spirit P2MP firmware applies them before IEEE802.1p and DSCP.

The following values are available to configure prioritization rules:

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

Table 21: Prioritization Rules Order Configuration

Prioritization Rules Order Parameter	Default Value
PACKETCLASSIFIER.GENERAL.RULES_ORDER	DSCP_VLAN

IEEE 802.1p Support

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

The Spirit P2MP firmware implements the mapping between IEEE 802.1p traffic classes and ITU-T G.hn traffic classes.

The following table lists the corresponding configuration parameters.

Table 22: IEEE 802.1p Prioritization Parameters

Parameter	Value
PACKETCLASSIFIER.GENERAL.VLAN_CLASS_MAP_EN	YES or NO Default = No
PACKETCLASSIFIER.GENERAL.VLAN_CLASS_MAP	Table where each position represents an 802.1p priority. The value associated to a position is the G.hn class for this 802.1p priority. Default values = 2,0,1,3,4,5,6,7

DSCP Support

The Differentiated Services (DiffServ) is a traffic management model which specifies a mechanism for classifying and managing network traffic, and providing quality of service (QoS) on IP networks.

The DiffServ architecture uses a 6-bit Differentiated Service Code Point (DSCP), which replaces the ToS field in IPv4. In the DS field, a range of eight values is used to enable backwards compatibility with IP precedence specifications in the outdated ToS field.

The Spirit P2MP firmware implements mapping between the DSCP and ITU-T G.hn classes.

The following table lists the corresponding parameters.

Table 23: DSCP Configuration

Parameter	Value
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, DSCP value 8 is associated to CLASS_MAP[8] value (equal to 1), thus DSCP value 8 is classified as class 1. 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	Enable or Disable the DSCP to G.hn class mapping. Default = YES

Custom Rules

Using the Spirit P2MP firmware, you can configure user-defined prioritization rules by delegating a pattern matching rule and a set of packet classification rules.

There are two pattern matching rules, which can be enabled independently.

A packet matching rule can be defined as follows:

- Offset: Offset inside the Ethernet packet where the bitmask and pattern should be applied. The offset is in 16-bit units. Offset 0 is the first byte of an Ethernet packet (the destination MAC address LSB).
- Bitmask: 16-bit bitmask can be applied to the 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 are matched.

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

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

Table 24: Custom Rules Parameters

Parameter	Value
PACKETCLASSIFIER.GENERAL.TYPE_CLASS_MAP_EN	Enables custom rules. If set to YES, custom defined rules are configured in the packet classification hardware. Default = NO
PACKETCLASSIFIER.GENERAL.TYPE_CLASS_MAP	Table containing the mapping between classification rules and priorities.
PACKETCLASSIFIER.GENERAL.MATCHING_RULES	Table containing the offset, bitmask, pattern, and enable tuple.
PACKETCLASSIFIER.GENERAL.CLASSIFY_RULES	Table containing the offset, bitmask, pattern, and enable tuple.
PACKETCLASSIFIER.GENERAL.FRAME_TYPE	Indicates whether the offset used in the rules are referred to Ethernet packets with or without 802.1Q.

Predefined Prioritizing Traffic

The Spirit P2MP firmware implements a set of predefined rules to enable the prioritization of commonly found traffic types 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. This improves the performance of TCP traffic.
- **ARP:** Frames used for address resolution. Prioritizing these packets guarantees that the ARP protocol continues working under congested environments.

The following table lists the corresponding configuration parameters.

Table 25: TCP ACK Prioritization Parameters

Parameter	Value
PACKETCLASSIFIER.GENERAL.TCPACKV4_CLASS_MAP_EN	YES or NO Default = YES
PACKETCLASSIFIER.GENERAL.TCPACKV4_CLASS_MAP	0-7 Default = 4
PACKETCLASSIFIER.GENERAL.TCPACKV6_CLASS_MAP_EN	YES or NO Default = YES
PACKETCLASSIFIER.GENERAL.TCPACKV6_CLASS_MAP	0-7 Default = 4
PACKETCLASSIFIER.GENERAL.ARP_CLASS_MAP_EN	YES or NO Default = YES
PACKETCLASSIFIER.GENERAL.ARP_CLASS_MAP	0-7 Default = 6

Application Features

Encryption

The Spirit P2MP firmware can encrypt data connections using a single encryption key. It uses the AES-128 encryption algorithm, as specified in *G.9961 subclause 9.1*, to allow for the use of a variable-size in the Message Integrity Code (MIC).

User Interface

GPIOs Configuration

The device contains a set of GPIOs that can be controlled from within the firmware to manage external hardware. The provided API enables the configuration, reading, writing, and triggering of interrupt requests (IRQs) on any of the available GPIOs.

If there is a button or an LED connected to a GPIO, additional APIs are available to facilitate the registering of a button that is pressed or to program LED states.

Buttons and LEDs

Buttons and LEDs can be configured as needed. An API is provided to customize the information that is used to update the state of the different LEDs connected to the device and the behavior of the buttons.

By default, the LEDs are updated with information about the quality of the connection and the activity of the Ethernet interface. Buttons can perform functions, such as a factory reset.

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

Configuration Layer

The Spirit P2MP firmware implements a configuration layer which contains all of the configuration parameters of the MaxLinear G.hn node. It is used by other firmware components to configure its operation mode and to report information.

The configuration layer provides a single abstraction layer between firmware components, user interface, and adapter management tools.

The configuration parameters can be accessed using the SCT or the web server that is embedded in the system.

Some parameters are only for *POST* operations (these are write-only), others are only for *GET* operations (these are read-only), and the rest are for *POST* and *GET* operations (these are read-write). All these parameters can be accessed on the **Advanced Configuration** tab on the SCT of MaxLinear G.hn devices.

Parameters that need to be present after a reset are stored in the flash file system. This file can be customized through the PCK. This kit provides a GUI tool so that you can encode 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 embedded in the tool delivered with the firmware.

LCMP

The Spirit P2MP firmware includes support for the Layer 2 Configuration Management Protocol (LCMP) and implements the data model that is required for the HomeGrid Forum Compliance and Interoperability (C&I) automation program.

A private data model has been defined by exposing the configuration layer. Access to this data model is available using the Device Embedding Kit (DEK).

Web Server

The embedded web server enables remote management and configuration of the device using a standard web browser. Static pages are stored in the flash file system in the **B:\web** folder.

These pages can be customized and uploaded to the modem during runtime using the One Step Upgrade Procedure (OSUP). The PCK can also be used to create customized flash images with custom web content.

Java-script is used to access modem information through the configuration layer.

> ***Note:*** The default web page is used as an example. Configuration of all the available parameters is not supported on it.

Log File

The Spirit P2MP firmware uses a log file to periodically store information from the node in a local file that can be uploaded to a remote FTP server. This enables users to monitor network operations remotely, and capture data that can be used for troubleshooting purposes.

The maximum size of the log file is 256Kbytes.

The files are uploaded to the FTP server using the following naming convention: Logfile, followed by the MAC address, followed by the sequence number. For example, *logfile_00b9d093ac23_24*.

Using the log files, you can monitor any parameters that are in the configuration layer. The parameters are then stored in a text file in the flash file system, located in **b:/logfile/logfile.cfb**.

The file can be updated using the OSUP.

Table 26: Example of a Configuration File in the Log File

Example
SYSTEM.MISC.UPTIME
DIDMNG.GENERAL.DIDS
DIDMNG.GENERAL.MACS
DIDMNG.GENERAL.TX_BPS
DIDMNG.GENERAL.RX_BPS
FLOWMONITOR.STATS.FEC_HISTOGRAM
MASTERSELECTION.STATS.DESC
MASTERSELECTION.STATS.INFO

The log file can be configured using the following parameters:

Table 27: Log File Parameters

Parameter	Value
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. Set to 0 to disable automatic uploads (in minutes).
LOGFILE.GENERAL.ENABLE	Specifies if the logging feature is enabled.
LOGFILE.GENERAL.SEND_FILE	If the value is, ■ Write: Log file is sent to the local disk or to the FTP server ■ Read: Returns the status of the send file progress. If: — 0 = Finished — 1 = Still in progress
LOGFILE.GENERAL.DESTINATION	Configure log file destination. ■ LOCAL: Log file is stored in the modem (In the following location: *b:/logfile/logfile_last*). ■ FTP: Log file is sent by the FTP to the server.
LOGFILE.GENERAL.STATUS	Report information about the log process. It displays 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.

Cable Wire Length Measurement

The Spirit P2MP firmware uses an algorithm to measure the length of the coaxial cable between the HE and each CPE. The measurement is based on the propagation delay, which is a characteristic of the cable type used. The measurement is represented as a percentage of the speed of the light propagation in the vacuum.

- In the HE, the wire length is reported for each CPE.
- In the CPEs, the wire length is only reported for the HE.

The distance is reported in meters with an error of ±5m.

Table 28: Cable Wire Length Parameters

Parameter	Value
DIDMNG.GENERAL.VELOCITY_FACTOR	The velocity factor in the medium is the propagation speed of the electrical signal as a percentage of the speed of light in the vacuum. It depends on the dielectric that is used. Units: Percentage of the speed of light in the vacuum. Default: 66 (RG-59 coax cable).
DIDMNG.GENERAL.WIRE_LENGTH	Table that contains the wire length, estimated in meters. Each position represents a node (DID). The correspondence between DID and MAC for each remote node can be checked in the DIDMNG.GENERAL.MACS table of the MACs. Every entry index corresponds to the DID for the remote node. In case of a: ■ HE: It contains all the CPEs registered in the network. ■ CPE: It only contains the HE.

MaxLinear, Inc.
5966 La Place Court, Suite 100
Carlsbad, CA 92008
760.692.0711 p.
760.444.8598 f.
www.maxlinear.com